

TÜRKMEN POLITEHNIKI INSTITUTY

O. Nurgeldiýew

C++ ulgamlaýyn maksatnama dili

Ýokary okuw mekdepleri üçin okuw kitaby

Aşgabat – 2010

O. Nurgeldiyew, C++ ulgamlaýyn maksatnama dili

Ýokary okuw mekdepleri üçin okuw kitaby, Aşgabat – 2010 ý.

Giriş

Kompýuter tehnologiýasy – iň ýaş ugurlaryň biridir. Kompýuter tehnologiýasynyň ösüş taryhy beýlekilere garaňda kän bir uly döwri alýan däldir – 40-50 ýyl. Kompýuter tehnologiýasy, tehnikasy diýen adalgalar bolsa ondan hem ýaşdyr. Bilişimiz ýaly, ilkiňbaşda elektron-hasaplaýyş maşyn, hasaplaýyş tehnikasy diýen adalgalar ulanylardy. 20 ýyldan bäri bolsa EHM diýlen adalga ýuwaş-ýuwaşdan ýitip kompýuter adalga öz ornuny berdi, tehnika bolsa öňküleri ýaly hasaplaýyş dälde kompýuter tehnikasy diýlip atlandyrylýar.

Kompýuter tehnologiýasy ýaş bolmak bilen, dünýäde öňdebaryjy ugurlaryň biri bolup durýar. Häzirki wagtda habarlar-aragatnaşyk tehnologiýalaryň ýokary depginde ösýändigini barada aýdylýar. Öýjükli telefon aragatnaşygyň, maglumat tehnologiýalaryň ösmegi muňa subut bolup durýar. Şol tehnologiýalaryň düzümine çuňlaýyn seredilen mahalynda olaryň kompýuter ugruna esaslanýandygyna göz ýetirmek bolýar.

Öz gezeginde, kompýuterler öz düzüminde tehnikanyň başga ugurlarynyň soňky derejelerini jemländir. Bu bolsa onuň bilen işlemegi diňe ýeňillemän, eýsem amatly edip goýýar.

Türkmenistan dünýä ösüşiniň gapdalynda durman, kompýuter tehnologiýalaryň soňky gazananlaryny ulanmaklyga ymtylýar. Ýurdumyzda öňdebaryjy tehnologiýalary öwrenmeklik boýunça uly işler alnyp barylýar. Şol işlerde Hormatly Prezidentimiziň ýardam bermegi olaryň tiz depginde amala aşmagyny üpjün edýär. Ýurdumyzyň Baştutany öz gymmatly wagtyny tygşytlanman dünýäniň ösüşindäki ymtylyşlara üns berýär we olaryň has netijelilerini döwletimizde gerekli ugurlarda ornaşdyrylmagyna ýardam berýär.

Täze galkynyşlar zamanasynda ýurdumyzyň islendik pudagyň önünde täze meseleler goýuldy. Şol meseleleri üstünlikli çözmek üçin diňe bir tehnologiýalar ýeterlikli däl. Şol tehnologiýalary ulanyp biljek ýokary derejeli hünärmenler zerur.

C - programma düzmekligiň umumy dilidir. Bu dil UNIX sistemasy bilen berk baglanyşyklydyr, ýagny bu sistemada döredilendir, öz gezeginde UNIX sistemasynda işleýän programmalaryň aglabasy, şeýle hem UNIX sistemasynyň özüniň soňky wersiýalary C dilinde ýazylandyr. Ýöne diliň özi anyk bir maşyn bilen ýa-da operasion sistema bilen aýrylmaz baglanyşykly däldir. Bu dil "sistemany programmirlemekligiň dili" diýip atlandyrylýar, sebäbi C dilinde kompilýatorlary we operasion sistemalary yazmak has amatly, şeýle hem bu dilde beýleki uly programma önümlerini döretmek üçin hem juda giň mümkinçilikler bar.

C dili döredilende köp wajyp ideýalar Martin Riçards tarapyndan döredilen BCPL dilinden alnypdyr. Ol täsir 1970-nji ýylda Ken Tompson tarapyndan PDP-7-de döredilen, UNIX-iň ilkinji sistemasy üçin niýetlenen B diliniň üsti bilen geçipdir.

BCPL we B "tipsiz" dillerdir. Olardan tapawutlylykda C dili maglumatlaryň dürli tipleri bilen üpjün edýär. Baza tipine harplar(char), şeýle hem dürli ölçegli bitin we hakyky sanlar girýär. Bulardan başgada görkezijilerden(pointer), massiwlerden(array), gurluşlardan(structure) we birleşmelerden(?) programmistiň özüne tipleriň giden toplumyny döretmäge mümkinçilik berilýär. Aňlatma

operatorlardan we operandlardan düzülýär. Islendik aňlatma, eýe bolmak we funksiýany çagyrmak aňlatmalaryndan başlap hemmisi instruksiýa(görkezme) bolup biler. Görkezijiler maşyna bagly bolmadyk adres arifmetikasy bilen üpjün edýär.

C-de gowy gurluşly(strukturaly) programmalarda ulanylýan, esasy dolandyryjy konstruksiýalar bar: düzme instruksiýa ({...}), şert boýunça şahalanmak(if-else), birnäçe ýoldan birini saýlap almak(switch), şerti başda barlanylýan gaýtalanmak operatory(while, for) we şerti soňynda barlanylýan gaýtalanmak operatory(do), şeýle hem gaýtalanmaklygyň arasyny üzýän serişde(break).

Programma ýazmagyň düzgüni

Function-lara(kömekçi programma) ýüzlenilenden soňra, olar netije görnüşinde baza tiplerini, structuralary, birleşmeleri we görkezijileri berip bilerler. Islendik function özüne rekursiw ýüzlenip biler. Düzgün bolşy ýaly, function-yň lokal üýtgeýän ululyklary funktion-a her gezek ýüzlenilende “awtomatiki”, ýagny gaýtadan döredilýär. Bir function-y beýleki function-yň içinde kesgitlep bolmaýär, ýöne üýtgeýän ululyklaryň deklarasiýasyny blok-struktura görnüşinde kesgitlemäge mümkinçilik berilýär. C dilinde ýazylan programmanyň function-lary aýratyn faýlda saklanyp we aýratynlykda kompilirlenip biliner. Function-lardaky üýtgeýän ululyklar içki we daşky üýtgeýän ululyklara bölünýärler. Daşky üýtgeýän ululyklar bir faýlyň çäginde ýa-da bütün programmada ulanylyp biliner.

Preprosessirlemek etapynda programmanyň tekstiniň içine daşyndan makroslar, beýleki programmalar(ishodnik) gaýulýar we şertleýin kompilirlenýär.

C dili beýleki köp dillere görä "pes derejeli" dil hasaplanýar. Ýöne munuň özi onuň abraýyny peseltmeýär, ol kompýuterleriň köpüsiniň iş salyşýan obýektleri bolan, ýagny harplar(char), sanlar, we adresler bilen işleýär. Olar bilen real maşynda bar bolan arifmetiki we logiki operasiýalary ulanmak arkaly iş salyşmaklyga mümkinçilik berýär.

C dilinde setirler(string), köplükler, spisoklar, massiwler ýaly düzme obýektleriň üstünden geçirilýän göni operasiýalar ýok. Bu dilde structure-lara(gurluş) bir obýekt hökmünde seredilip olaryň tutuş kopiýasyny almak bolýan hem bolsa, bitin tipli massiwler ýa-da setir ululyklary bilen işlemäge mümkinçilik berýän operasiýalar ýok. Funksiýanyň lokal üýtgeýän ululyklary üçin ýer bölünip berlende ulanylýan static üýtgeýän ululyklar we stack mehanizminden başga kompýuteriň ýadyny paýlamaga mümkinçilik berýän serişde ýok. Şeýle hem dürli maglumatlary ýerleşdirmek üçin ulanylýan "heap-ýady"(памяти-кучи) we "hapany ýygnaýjy"("сборщик мусора") ýok. C diliniň özünde giriş-çykyş serişdesi, ýagny READ(okamak) we WRITE(ýazmak) instruksiýalary ýa-da faýllara ýüzlenmekligiň başga bir usuly ýok. Bularyň hemmesi – ýokary derejeli mehanizm bolup, C dilinde funksiýalary aýyk çagyrmak arkaly amala aşyrylmalydyr. C ulgamynyň(diliniň) köp görnüşinde gerek bolýjak şeýle function-laryň toplumy bar.

Ýokardaky aýdylanlary dowam etmek bilen belläp geçmeli zat, C dili hasaplamany gidişini diňe yzygiderli dolandyrmaga mümkinçilik berýän serişde bilen üpjün edýär: şerte görä şahalanmak mehanizmi, gaýtalanmak, düzme instruksiýa, kömekçi programma. Multiprogrammirlemek, parallel prosesler, sinhron we programmany gurnamak serişdelerini bolsa özünde saklamalýar.

Ýokardaky agzalyp geçilen serişdeleriň bolmazlygy uly kemçilik ýaly bolup görünýär(“iki setiri deňeşdirmek üçin function-a ýüzlenmeli bolýar”). Ýöne diliň gysgalygynyň(ykjamlygynyň, kompaktlygy) hakyky peýdasy iş ýüzünde ýüze çykýar. Umuman aýdylanda C dili beýleki diller bilen deňeşdureninde göwrümi boýunça az, şeýlelikde onuň düşündirilişi hem gysga, şonuň üçin ony beýleki dillere garanynda az wagtda öwrenmek bolar. Özünem programmist şol az wagtda C diliniň hemme mümkinçiliklerini öwrenjekligine, düşünjekligine we praktikada ulanyp biljekligine bil baplap biler.

Köp ýyllaryň dowamynda “C programmirlemek dili” atly kitabyň, birinji neşiri bu dil baradaky ýeke-täk standart(kesgitleme, kitap) bolup hyzmat edip geldi. 1983-nji ýylda amerikanyň milli standartlar instituty(ANSI) tarapyndan C diliniň döwrebap standartyny(kesgitlemesini) taýýarlamak boýuça komitet döredilýär. Ol komitet tarapyndan 1988-nji ýylda C diliniň standarty(“ANSI-C”) taýýarlanylýar. Ol standartyň köp düzgüni häzirki zaman kompilýatorlarynda ulanylýar.

Standart ilkinji çykan sorag-jogap gollanmasyna esaslanýar. Ol gollanma bilen deňeşdirilende dil az özgeripdir. Standartyň esasy maksatlarynyň biri, bar bolan programmalaryň köp böleginiň kabul edilen standarta gabat gelmegidir ýa-da ol programmalaryň standarta gabat gelmeýän ýerleri barada kompilýatoryň duýduryjy habarlary bermegidir.

Programmistleriň köpüsi üçin function-y beýan etmegiň we kesgitlemegiň täze sintaksisi kabul edilen standartdaky in wajyp täzelik boldy. Indi function bilen birlikde onuň argumentleri hem kesgitlenip biliner. Şeýlelikde function-y kesgitlemekligiň sintaksisi hem üýtgedi. Argumentleriň kesgitlenmegi bilen ýüze çykýan goşmaça maglumat, tipleri gabat gelmeýän argumentler bilen baglanyşykly näsazlyklary tapmakda kompilýatoryň işini has ýeňilleşdirýär; munuň özi biziň pikirimizçe dile edilen örän peýdalý goşundydyr.

Ýene-de uly bolmadyk täzelikleriň bir toparyny belläp geçmek artykmaçlyk etmez. Dildäki esli wagt bäri giňden ulanylyp gelinýän structure we enaum(?) tipi düzgünleşdirildi. Indi süýşýän nokat bilen hasaplamak birlik takyklykda hem ulanylyp biliner. Umuman arifmetikanyň häsiýeti, esasan hem alamatsyz tipler üçin aýdyňlaşdyryldy. Preprocessor kämilleşdirildi. Programmistleriň köpüsi bu üýtgeşmeleri örän gowşak ulanýarlar.

Standartyň ikinji bir aýdyp geçmeli esasy tarapy, operasion sistemanyň mümkinçiliklerini ulanmaga ygtyýar berýän(mysal üçin faýllary okamaga we ýazmak), formatly giriş-çykyş, ýady dinamiki ulanmaklyk, string(literlerden düzülen zynjyrlyk) bilen işlemeklik üçin gerek bolan, C kompilýator bilen goýulýan bibliotekalaryň kesgitlenmegidir. Standart baş faýllaryň toplumynyň bolmagy, maglumatlaryň tipleriniň we functiony beýan etmekligiň birmeňzeş usuly bilen üpjün edýär. Bu bibliotekalary ulanýan programmalar şol operasion sistemany ulananda, islendik başga maşynlarda hen ýerine ýetjekdigine güwä geçilýär. Bibliotekany düzýän köp programmalar UNIX sistemasynyň "standart giriş-çykyş biblioteka"-synyň nusgasynda we meňzeşlikde döredilendir. Bu biblioteka kitabyň birinji neşirinde ýazylyp görkezilýär we beýleki sistemalarda hem giňden ulanylýar. Bu ýerde hem programmistler düýpli tapawudy duýmazlar.

C diliniň maglumatlarynyň tipi we dolandyryjy düzümi(strukturasy) bar bolan maşynlaryň köpüsiniň komandalary bilen gollanylýar, programmanyň özbaşdak göyberilmeginiň we hasalama geçirilmeginiň administratiw sistemasy bolsa örän seýrek duşýar. Bibliotekadaky functionlara ýüzlenmekligi programmist ýazýar(kompilýator däl), şonuň üçin hem isleg bolanda olary beýlekilere çalyşmak bolar. Operasion sistemadaky käbir gizlin detallar bilen bagly aýratynlyklary ulanmaýan C dilinde ýazylan programmalaryň hemmesi diýen ýaly, islendik maşynda ýerine ýetýär.

C dili belli bir, konkret maşynyň arhitekturasy göz önünde tutulyp ýazylan däl, ýöne ol köp maşynlaryň apparat mümkinçiliklerine gabat gelýär. Belli bir kadalara eýermek bilen, islendik maşynda ýerine ýetýän C programmany ýazmak bolar. Standart, programmanyň ýerine ýetjek maşynynyň karakteristikalaryny görkezýän hemişelikleriň toplumyny ulanmak bilen, islendik maşynda ol programmanyň ýerine ýetmekligini gazanmaklyga mümkinçilik döredýär.

C "berk tipleşdirilen" dil bolmasa hem, ol öz ösüş döwründe tipleriň üstünden barlagy güýçlendirdi. C diliniň ilkinji wersiýalarynda, bitin sanlar bilen görkezijileriň arasynda barlagsyz çalyşmaklyga rugsat edilýärdi, munu özi köp düşnüksizlige sebäp bolýardy, şonun üçin hem C diliniň soňky wersiýalarynda beýle zat gadagan edildi. Standarta baglylykda, täze kompilýatorlar ulanylýan maglumatyň tipini we tipleriň özgerdilmesini aýyk görkezmekligi berk talap edýär. Functiony beýan etmekligiň täze görnüsi - bu tarapa ädilen ýene bir ädimdir. Indi kompilýator tiplerdäki ýalňyşlyklaryň köp bölegi barada duýdurýş berýär we ylalaşmaýan tipleri awtomatiki özgertmekligi amala aşyрмаýar. Programmistler öz etjek işini bilýändir - diýen C-diliniň esasy pelsepesi öz güýjünde galyp, diňe meseläniň çözülmegi üçin has anyklaşdyrmaklygy talap edýär.

C dili hem beýleki diller ýaly kemçiliklerden halas däl. Käbir operatorlaryň ululyk derejesi umumy kabul edilen däl, käbir sintaksis gurluşlar has hem gowy edilip gurnalyp bilinerdi. Şeýlede bolsa C dili, köp meseleleri çözmek üçin amatly, adatdan daşary effektiv we düşnükli dildir.

Kitabyň şeýle düzümi bar. 1-nji bab C diliniň esasy serişdelerine syn. Ol okyjynyň mümkin boldugyndan çalt programma ýazmaklygyna höwes döreder, ýagny biziň ynanyşmyza görä täze dili öwrenmekligiň ýeke täk usuly ol dilde programma ýazmakdyr. Kitabyň bu bölegine oňat düşünmek üçin, programmirlemekligiň esasy elementlerinden habarly bolunsa has oňat bolar. Bu ýerde kompýuter näme, kompilýasiýa näme ýa-da $n=n+1$ aňlatmanyň nämäni aňlatýanlygy barada hiç-hili düşündiriş berilmeýär. Mümkin bolan ýerde programmirlemekligiň peýdaly usullaryny görkezmäge çalyşan hem bolsak, bu kitabyň maglumatlaryň düzümi boýunça-da, algoritmler boýunça-da sorag-jogap gollanmasy bolmaklygy göz-önünde tutulmady; bu ýerde esasan nämä ünsi çekilýär diýilende bolsa, diliň özüne ünsi çekilýär - diýilse has dogry bolar.

2-6-njy bablarynda bolsa diliň dürli serişdelerine 1-nji bapdaka garanda has çintgiläp we has formal(umumy) seredilip geçilýär; özünem bu ýerde hem edil öňki ýaly izolirlenen böleklere dälde, gutarnykly programmalar bolan mysallara ünsi çekilýär. 2-nji bapda maglumatlaryň baza tipi, operatorlar we aňlatmalar bilen tanyşdyrylýar. 3-nji bapda hasaplamaklygyň yzygiderligini dolandyrmaklygyň serişdelerine seredilip geçilýär: if-else, switch, while, for we ş.m. 4-nji bapda function we programmanyň düzümine(daşky üýtgeýän ululyklar, üýtgeýän ululyklaryň görüniş düzgüni, programmany birnäçe faýllara bölmeklige e ş.m.), şeýle hem preprocessor barada gürrüň berilýär. 5-nji bapda görkezijiler we adres arifmetikasy meselesine seredilip geçilýär. 6-njy bap bolsa structura we birleşmä bagşylanandyr.

7-nji bapda, operasion sistema bilen umumy interfeýsi üpjün edýän standart biblioteka barada gürrüň berilýär. Bu biblioteka ANSI standartyna girizilipdir,

başgaça aýdanymyzda bu biblioteka C dili bar bolan maşynlaryň ählisiniň düzümine girýär, şoňa göräde operasion sistemanyň giriş-çykyşyny we beýleki mümkinçiliklerini ulanýan programmany hiç-hili üýtgetmezden bir maşyndan beýleki maşyna geçirmek bolar.

8-nji bapda C dilinde ýazylan programmalar bilen UNIX operasion sistemasynyň arasyndaky interfeýsi, esasan hem giriş-çykyş, faýl sistemasy we ýadyň paýlanyşy baradaky mesele gozgalýar. Bu babyň birnäçe bölümi UNIX sistemasynyň aýratynlyklaryny bilen baglanyşykly bolsa-da, beýleki sistemalarda işleýän programmistler hem bu ýerde standart bibliotekanyň döredilişi barada, ýazylan programmanyň dürli maşynlarda ýerine ýetmegi üçin näme etmelidigi barada köp peýdaly maglumatlary taparlar.

Kitabyň soňunda berilýän A goşmaça sorag-jogap bölümçesi hökmünde seretmek bolar. C diliniň sintaksisiniň we semantikasynyň berk kesgitlemesi ANSI standartynyň ýörite dokümentinde saklaýar. Bu dokument esasan hem kompilyatory döredijiler üçin has hem zerurdyr. Biziň sorag-jogap bölümçämiz standartyň ulanýan kanunalaýyk stillerine köp üns bermän dile gysgajyk kesgitleme berýär. B goşmaça - standart bibliotekalar barada maglumatlary özünde saklaýar we ol kompilyator döredijiler üçin däl-de esasan ulanyjylara(programmaýsler) niýetlenendir. C goşmaça bolsa diliň birinji wersiýasyndakydan tapawudynyň gysgaça beýanyňy özünde saklaýar. Ýöne düşnüksiz ýagdaýlarda dil boýunça gutarnykly netije çykarýan emin, diliň standarty we siziň ulanýan kompilyatoryňyz bolup galýar.

Turbo c++ diliniň mümkinçiliklerine syn

C dili bilen gysgaça tanyşmakdan başlalyň. Biziň maksadymyz - ownuk-uşak zatlar, formal düzgünlere üns bermezden real programmalaryň üsti bilen diliň elementlerini açyp görkezmekden ybarat. Şonuň üçin hem biz dolylyga hem-de takyklyga ymtylmaýarys(ýöne mysallaryň korrektligi barada alada etdik). Eýle çemeleşmek bilen biz siziň peýdaly programmalary ýazjak wagtyňyzy ýakynlaşdyrmak isledik. Şonuň üçin siziň ünsiňizy esasy zatlar: üýtgeýän ululyklara, hemişeliklere, arifmetika, hasaplamaklygyň yzygiderligini dolandyrmaklyga, funksiýalara we ýonekeý giriş-çykyşa jemlemek zerurdyr. Bu bapda uly programmalary ýazmak üçin gerek bolan serişdelere seretmeris. Olar görkezijiler, structure-ler, operatorlaryň köp bölegi, birnäçe dolandyryjy görkezmeler(instruksiýa) we standart bibliotekalar.

Beýle çemeleşmekligiň hem öz ýetmezçiligi bar. Olaryň biri hem biz bu ýerde diliň häsiýetini gutarnykly beýan edilmeýäris bu bolsa öz gezeginde käbir ýagdaýlara nädogry düşünmeklige getirip biler. Berlen materiallaryň çäklidigi üçin, biz seredip geçen mysallarymyzda diliň doly kuwwadyny ulanyp bilmedik, ýogsam olar has gysga hem gelşikli ýazylyp bilinerdi. Biz başardygymyzdan bu ýetmezçiligi duýulmaz ýaly edip bermeklige çalyşdyk, ýöne bu bapda ýazjak programmalarymyzda şeýle ýetmezçiligiň bardygyny aýdyp geçmek gerek. Beýleki bir ýetmezçilik, geljekgi baplarda käbir zatlary gaýtalap geçmeli bolarys.

Ýöne gaýtalamakdan geljek peýda, geljekde oňa sarp eden wagtymyza ödär diýip umyt edýäris.

Islendik ýagdaýda hem tejribeli programmist bu babyň materiallaryndan öz düzýän programmalarynda ulanar ýaly peýdaly zatlary tapyp biler. Öwrenje programmistler bolsa, diňe bu ýerde ýazylyp görkezilen programmalary barlap görmek bilen çäklenmän, özlari hem birnäçe kiçiräjik programmalary ýazyp görmek maslahat berilýär. Tejribeli programmistler hem, öwrenje programmistler hem bu babyň materiallaryna ullakan sütünler hökmünde seretmelidirler, 2-nji bapdan başlap ol sütünlerde diliň elementlerini asylyp goýuljakdyr.

Programmirlemek dilini öwrenmekligiň ýeke täk usuly - munuň özi ol dilde programma ýazmakdan ybaratdyr. Islendik dil öwrenilende-de düzgün bolşy ýaly ilki bilen takmynan aşakdaky ýaly programma düzmeklikden başlaýarlar:

Salam ! sözünü çapa çykarýan programma ýazmaly.

Ine ilkinji kynçylyklar: haýsydyr bir tekst redaktorynda programmanyň tekstini taýýarlamaly, ony kompilirlemeli soňra bolsa ýerine ýetirmeli we netijesiniň niräk çykjaklygyny kesgitlemeli. Bu zatlary öwreneniňizden soň, galan zatlar bize aňsat ýaly bolup görner.

Aşakdaky programma “Salam!” sözünü çap edýär:

```
#include <stdio.h>
main()
{
    printf(“Salam!\n”);
}
```

Bu programmany işe göýbermek siziň ulanýan sistemaňyza bagly. UNIX operasion sistemasynda programmany “.c” simwol bilen gutarýan faýlda ýerleşdirýärler, mysal üçin hello.c. Soňra bu faýly

cc hello.c

komandanyň kömegi bilen kompilirleýärler. Egerde siz belgileriň hemmisini ýerbe-ýer goýan bolsaňyz, orfografik ýalňyşlyk goýbermän hemme zatlary dogry ýerine ýetiren bolsaňyz, onda kompilirlemeklik “sessiz-üýnsiz” geçer. Netije-de ýerine ýetirmeklige taýyn a.out atly faýly alarsyňyz. Indi siz ol faýly

a.out

komandasynyň kömegi bilen ýerine ýetirseniz, programma

salam!

sözünü çap eder.

Beýleki sistemalarda programmany işe goýbermekligiň tertibi başgaça bolup biler. Ol barada degişli hünärmenler bilen maslahatlaşmaly.

Indi programmanyň özüne degişli käbir pursatlary aýdyňlaşdyrallyň. C dilinde ýazylan programma özüniň tutýan göwrümüne bagly bolmazdan function-lardan we üýtgeýän ululyklardan ybarat bolýar. Function-lar hasaplamak üçin gerek bolan görkezmeleri(komandalary, instruksiýalary) özünde saklaýan bolsa, üýtgeýän ululyklar hasaplama döwründe ulanylyan bahalary özünde saklaýarlar.

C dilindäki function-lar FORTRAN dilindäki kömekçi programmalara we function-lara, şeýle hem PASCAL dilindäki function-lara we Procedure-lere meňzeşdir. Ýokardaky görkezilen programma – munuň özi **main** atly functiondyr.

Adatça siž öz ýazan function-laryňyza islendik at dakyp bilersiňiz, ýöne **main** – aýratyn at bolup, islendik programma şol function-ynyň ilkinji görkezmelerini ýerine ýetirmek bilen öz işine başlaýar.

Adatça **main** öz işini ýerine ýetirmek üçin beýleki function-laryň hyzmatyndan peýdalanýar; olaryň käbirlerini programmistiň özi döredýär, beýlekileri bolsa onuň ygtyýarynda bolan gordan(library, biblioteka) alynýar. Programmanyň ilkinji setiri:

```
#include <stdio.h>
```

kompilýatora standart giriş-çykyş goryny birleşdirmelidigi hakynda görkezme berýär. Bu setir köp **C** programmanyň asyl nusgasyny saklaýan faýlyň başynda gabat gelýär. Standart(adaty) gor 7-nji bapda we B şoşmaçada beýan edilýär.

Function-laryň arasyndaky maglumat geçirmekligiň bir usuly hem bir function beýlekä ýüzlenende argument diýilip atlandyrylýan birnäçe bahalaryň toplumyny geçirmegidir. Ol ululyklaryň aralary otur bilen bölünip ýaý şekilli skobkanyň içinde, function-yň adynyň yzynda ýerleşdirilýär. Biziň mysalymyzda **main** function görnüşinde kesgitlenýär, bu ýerde ýaý şekilli skobkanyň içiniň boşlugy () onuň hiç-hili argumente garaşmaýanlygyny aňladýar.

C dilinde ilkinji programma

```
#include <stdio.h>
```

```
main()
```

```
{
```

```
    printf("Hello, World\n");
```

```
}
```

Bu ýerde:

#include <stdio.h> setiri, standart bibliotekany programmamyza birleşdirýänligimizi aňladýar.

main() hiç-hili argumenti bolmadyk main atly function-y kesgitleýäris. Bu function-yn görkezmeleri(komandalary) figuraly ýaýlaryň içinde ýerleşdirilýär.

printf("Hello, World\n"); main function-y simwollaryň berlen yzygiderligini çapa çykarmak üçin birleşdirilen bibliotekanyň **printf** function-yna ýüzlenýär. \n täze-setire geçmekligi görkezýän litera(simwol).

Function-yň komandalary figuraly ýaýlaryň içinde ýerleşdirilýär. Bu ýerde main functiony diňe bir görkezmäni özünde saklaýar:

```
    printf("Hello, World\n");
```

Function ady boýunça çagyrylýar we adynyň yzynda ýaý şekilli skobkanyň içinde argumentleriň sanawy görkezilýär. Şeýlelikde bu setirde **"Hello, World\n"** argument bilen **printf** functiony çagyrylýar. **printf** – biblioteka functiony bolup ol häzirki ýagdaýda goşa dyrnagyň içinde ýerleşdirilen simwollary çapa çykarýar.

"Hello, World\n" görnüşdäki goşadyrnagyň içinde ýerleşdirilen simwollariň yzygiderligine *string*(setir) ýa-da *string hemişeligi* diýilýär. Belli bir wagtlap printf we beýleki functionynlaryň argumenti görnüşinde diňe stringleri ulanarys.

C-de \n literleriň birleşmesi stringiň içinde täze setir literasyny aňladýar, ol çapa çykaryllanda indiki setiriň çep çäğine geçmekligi amala aşyrýar. Eger siz \n literany aýyrsaňyz(tejribede barlap bilersiňiz), onda çapa çykarmadan soň täze setire geçilmeýändigine göz ýetirersiňiz. Setir argumentinde täze setir(\n) literasynda aýyk görkezmeli. Mysal üçin siz aşakdaky ýaly ýazmak bilen:

```
printf("Hello, World  
");
```

indiki setire geçmekçi bolsaňyz onda kompilýator ýalňyşlygyň bardygy barada habar berer.

Täze setir literasy hiç-haçaň awtomatiki goýulmaýar, şonuň üçin bir setiri printf functionyna birnäçe gezek ýüzlenmek bilen hem çapa çykarmak bolar: Biziň ilkinji programmamyzy aşakdaky ýaly hem ýazmak bolar:

```
#include <stdio.h>
```

```
main()
```

```
{
```

```
    printf("Hello, ");
```

```
    printf("World");
```

```
    printf("\n");
```

```
}
```

Bu programmany ýerine ýetirsek öňki programmany ýerine ýetirenimizdäki setir çapa çykar.

\n diňe bir literany aňladýanlygyny belläliň. Esc yzygiderlik diýilip atlandyrylýn, \n ýaly ýapgyt çyzyk bilen başlanýan literalaryň aýratyn birleşmesi, göz önüne getirmesi kyn ýa-da görünmeýän literalary bellemek üçin giňden ulanylýar. Beýleki literalar bilen birlikde ulanylýan \t, \b, \", \\ literalar degişlilikde tabulýasiýany, bir litera yza gaýtmak(backspace), goşa dyrnak, ýapgyt çyzygyň özüni aňladýar. Şeýle literalaryň doly sanawy 2.3 bölümde görkezilendir.

Ýumuş 1. Öz sistemaňyzda **"Hello, World"** setiri ekrana çykarýan programmany ýerine ýetiriň. Programmanyň käbir böleklerini aýyrmak bilen dürli tejribeleri geçirip görüň, ýalňyşlyklar barada berilýän habara üňs beriň.

Ýumuş 2. printf functionyň setir argumentine ýokarda görkezilen sanawda bolmadyk \c literany goýsak näme bolýandygyny barlap görüň(bu ýerde c ýokardaky sanawda bolmadyk litera).

Üýtgeýän ululyk we arifmetiki aňlatma

Indiki programma $^{\circ}\text{C} = (5/9)(^{\circ}\text{F} - 32)$ formula boýunça hasaplamany geçirýär we aşakdaky temperaturanyň Ferengeýt we Selsiý boýunça gabat gelmek tablisasyny çapa çykarýar:

0	-17
20	-6
40	4
60	15
80	26
100	37

120	48
140	60
160	71
180	82
200	93
220	104
240	115
260	126
280	137
300	148

Öňki ýazan programmamyz ýaly bu programma hem ýeketäk main() functiondan ybarat. Bu programma “Hello, World” setiri çap edýän prigrammadan uzyn, ýöne kyn däl. Biz bu programmada kommentariýalary (düşündirişleri), üýtgeýän ululyklary beýan etmekligi, arifmetiki aňlatmalaryň ýazylyşyny, gaýtalanmaklygy we formatly çap etmekligi görkezeris:

```
#include <stdio.h>
/* fahr = 0, 20, ..., 300 üçin temperaturalaryň Ferengeýt we Selsiý boýunça
tablisyny çap etmek */
main()
{
    int fahr, celsius;
    int    lower, upper, step;

    lower = 0;          /* temperatyralaryň tabl. aşaky çägi */
    upper = 300;        /* ýokarky çäk */
    step = 20;          /* ädim */

    fahr = lower;
    while (fahr <= upper) {
        celsius = 5 * (fahr-32) / 9;
        printf("%d\t%d\n", fahr, celsius);
        fahr = fahr + step;
    }
}
```

/* fahr = 0, 20, ..., 300 üçin temperaturalaryň Ferengeýt we Selsiý boýunça tablisyny çap etmek */

bu iki setir *kommentariýa*(düşündiriş) bolup olar programmanyň ýerine ýetirýän işini gysgaça düşündirýär. /*, */ belgileriň arasynada ýerleşdirilen literalar kompilyator tarapyndan seredilmeýär we programmanyň düşnükli bolmagy üçin erkin ulanylyp biliner. Kommentariýa programmanyň islendik ýerinde ýerleşip, ol özünde boşluk literasyny, tabulýasiýa we täze setir literasyny saklap biler.

C dilinde islendik üýtgeýän ululyk ulanylmazyndan öň onuň haýsy tipe degişlidigi ýazylyp beýan edilmelidir; adaty hemme üýtgeýän ululyklar

functionyň başynda ilkinji ýerine ýetýän instruksiýadan öň beýan edilýär. Deklarasiýada(beýan edilişde) üýtgeýän ululygyň hasiýeti aýan edilýär. Ol tipiň adyndan we üýtgeýän ululyklaryň toplumyndan(spisogyndan) ybaratdyr, mysal üçin:

Int fahr, celsius;

Int lower, upper, step;

Int tipi görkezilen üýtgeýän ululyklaryň bitin sanly ululyklardygyny aňladýar, ondan tapawuplykda **float** tipi ýüzýän nokatly üýtgeýän ululyklary görkezýär, ýagny drob bölegi bolup biljek sanlary görkezýär. **Int** hem-de **float** tipiň alyp biljek bahalarynyň diapazony ulanylýan maşyna bagly.

Int tipi 16-razrýadly(ol -32768-den +32767-ä çenli diapazonda ýatýar), şeýle hem 32-razrýadly bolup bilýär. **float** tipli sanlar adaty 32-razrýadly sözler bilen ýazylýar, azyndan 6 sany manyly onluk sifri bolýar, 10^{-38} -den 10^{+38} -çenli diapazonda ýatýar.

C dilinde maglumatlar üçin **int** we **float** tiplerden başgaca birnäçe baza tipleri ulanmaklyga mümkinçilik berýär, olar:

char 1 - baýtly litera

short gysga bitin

long uzun bitin

double ikeldilen takyklykdaky ýüzýän nokatly

Görkezilen tipleriň obýektleriniň ölçegleri hem ulanulýan maşyna baglydyr. Baza tiplerden: ***massiwleri***, ***strukturalary*** we ***birleşmeleri*** (объединение), netije görnüşinde ol tipleriň bahalaryny yzyna gaýtarýan baza tiplerinden bolan obýektlere we functionlara ***görkezijileri*** döretmek bolýar. Olar hakynda soňra gürrüň ederis.

Temperaturany özgertmek programmasyndaky hasaplamak aşakdaky eýe bolmak instruksiýalaryndan başlanýar:

lower = 0;

upper = 0;

step = 20;

fahr = lower;

olar atlary görkezilen üýtgeýän ululyklaryň başlangyç bahasyny kesgitleýär. Islendik instruksiýa nokatly otur bilen gutarýar.

Tablisanyň hemme setirleri şol bir usul bilen hasaplanýar, şonuň üçin her bir setir üçin hasaplamany gaýtalanmaklygy ulanyp ýerine ýetireliň. while gaýtalanmasynyň ýerine-ýetmeginiň zerur şerti:

while (fahr <=upper) {

...

}

Ol şeýle ýerine ýetýär. Skobkalaryň içindäki şert barlanylýar. Eger ol çyn(true) bolsa(fahr ululygyň bahasy upper ululygyň bahasyndan kiçi ýada deň bolsa), onda gaýtalanmaklygyň göwresi(figuraly skobkalaryň içinde ýazylan üç instruksiýa) ýerine ýetýär. Soňra ýene-de şert barlanylýar we ol çyn bolsa gaýtalanmaklygyň göwresi ýenede ýerine ýetýär. Haçanda şert ýalan(false, ýagny fahr upperden uly bolanda) gaýtalanmak tamamlanylýar we hasaplamak gaýtalanmaklygyň yzyndaky

instruksiýadan başlanýar. Biziň musalymyzda görnüşi ýaly gaýtalanmaklykdan soň hiç-hili instruksiýa ýok, şonuň üçin programma öz işini tamamlayar.

while instruksiýasynyň göwresi bir ýa-da temperaturany özgertmek programmasyndaky ýaly figuraly skobkanyň içinde ýazylan birnäçe instruksiýadan, ýa-da aşakdaky ýalyskobkasyz ýazylan ýeke-täk instruksiýadan ybarat bolup biler:

```
while (i<j)
    i=2*i;
```

Ol ýa-da beýleki ýagdaýda hem biz while gaýtalanmaklygynyň dolandyryňan instruksiýalaryny bir tabulýasiýa süýşürüp ýazarys, ol bolsa programmada 4 sany boşluga deňdir(пробел); şonuň üçin hem gaýtalanmaklygyň göwresini düzýän instruksiýalar aýyl-saýyl görünýärler. Boşluklar programmanyň logiki düzümini kesgitleýär. C kompilýator programmanyň daşky oformleniýasyna üns bermeýär, ýöne gerek ýerinde goýlan tabulýasiýalar we boşluklar programma seredenimizde onuň düşnükli bolmagyna ýardam edýär. Aňlatmalaryň logiki düzüminiň görünmegi üçin olary aýratyn setirde we operasiýa bellikleri boşluklar bilen gurşamagy maslahat berýäris. Skobkalaryň ýerleşşi esasy däl, bu barada dürli pikirler bar. Biz olaryň birnäçe görnüşleriniň ulanylyşyndan biriniň üstünde durup geçdik. Olaryň içinden siz öz halanyňyzy saýlap alyň we şonuň bilen işlemegi dowam etdiriň.

Hasaplamanýň köp bölegi gaýtalanmaklygyň göwresinde ýerine ýetirilýär. Temperatura

$$\text{celsius} = 5 * (\text{fahr} - 32) / 9;$$

instruksiýanyň kömegi bilen Selsiý şkalasyna geçirilýär we celsius üýtgeýän ululygyna dakylýar. Biz bada-bat 9/5-e köpeltmän, ilki 5 –e köpeldýäris we soňra 9-a bölýäris. Beýle etmegimiziň sebäbi başgada köp dillerde bolşy ýaly C dilinde hem bitin sanlar bölünende kesilmegi bilen, ýagny drob böleginiň taşlanýanlygy bilen baglanyşyklysy. Şeýlelikde 5 we 9 bitin sanlar, kesilme netijesinde 9/5 aňlatma nul bahany kabul eder, netijede Selsiý boýunça temperaturanyň ýerine nul baha çap edilirdi.

Bu mysal printf – iň işleýşi barada hem düşüňjämizi giňeltdi. printf – munuň özi formatly giriş-çykyşyň uniwersal funksiýasydyr, ol barada 7-nji başda giňişleýin beýan ediljekdir. Onuň birinji argumenti string, ondaky her bir % bellik bolsa indiki argumentler(ikinli, üçünji) bilen baglanyşykly bolup, % belgisiniň yzyndan gelýan maglumat bolsa olaryň her-biriniň çapa çykjak görnüşini kesgitleýär. Mysal üçin %d argumentiň bitin san görnüşde çapa çykarylmany gurnaýar,

```
printf(“%d\t%d\n”,fahr,celsius);
```

instruksiýa bolsa bitin fahr ululygy çapa çykaryýar, tabulýa goýýar we bitin celsius ululugu çapa çykaryýar.

printf functionynda birinji argumentiň her bir spesifikatoryna(% bilen başlanýan gurluş) ikinji, üçünji, we ş. m. gabat gelýär. Spesifikatorlar we olara degişli argumentler sany we tipi boýunça gabat gelmelidirler: eger gabat gelmese onda maglumat nädogry görnüşde çapa çykarylar.

Ýeri gelende aýtsak printf C diliniň bölegi däl, umuman C dilinde giriş-çykyşa kesgitleýän ýörite konstruksiýa ýokdur. printf functiony C programma elýeterli bolan standart bibliotekadaky peýdaly functiondyr. printf functionynyň häsiýeti ANSI standart tarapyndan kesgitlenendir we standarta tabyn bolan hemme C sistemalarda birmeňzeşdir.

C-iň özüne ünsüňizi jemlemek üçin giriş-çykyşa barada 7-nji baba çenli köp gürrüň etmeris. Şeýle hem ady agzalan baba çenli formatly giriş barada-da gürrüň etmeris. Egerde sizde sanlary girizmek zerurlygy çyksa, onda **scanf** functionyna deňişli 7.4 bölümi okamaklygy maslahat berýäris. Bu functionyň ýazylyşynda printf bilen meňzeşlik bar, ýöne ol maglumatlary çapa çykarmaga däl-de girizmekligi üpjün edýär.

Temperaturany özgertmek programmasy bilen baglanyşykly ýene iki sany düzedilmeli mesele bar. Olaryň birinjisi(has ýönekeýi) çapa çylarylýan sanlar gysyk, ýagny olar kolonkanyň sag pozisiýasy boýunça deňlenmändir. Onu düzetmek üçin, her bir %d spesifikatora oňa berilýän meýdançanyň uzynlygyny görkezýän formaty goşmak ýeterlikdir; Şondan soňra programma sanlary görkezilen meýdançanyň sag tarapyna gysmak bilen çapa çykarar. Mysal üçin

```
printf("%3d %6\n", fahr, celsius);
```

görnüşde ýazsak, onda her bir setirde birinji san üç pozisiýada, ikinji san bolsa alty pozisiýada ýerleşdiriler. Netijede aşakdaky ýaly görnüşde çapa çykarylar:

```
0      -17
20     -6
40      4
60     15
80     26
100    37
```

...

Düzedilmeli meseleleriň ikinji has zerury biz bitin sanly arifmetikany ulanp hasaplama geçirdik, şonuň üçin hem Selsiý boýunça temperaturanyň tablisasy beýle bir takyk däl. Mysal üçin 0°F hakykatda(onluk takyklykda) -17,8°C-a dälde -17-ä deň. Has takyk temperaturanyň bahasyny hasaplamamak ucun biz bitin sanly däl-de eýsem ýüzýän nokatly arifmetikadan peýdalanmaladyrys. Bu bolsa programmada birnäçe üýtgetmäni geçirmekligi talap edýär.

```
#include <stdio.h>
```

```
/* fahr = 0, 20, ..., 300 üçin temperaturalaryň Ferengeýt we Selsiý  
boýunça tablisyny çap etmek */
```

```
main()
```

```
{
```

```
    float fahr, fahr, celsius;
```

```
    int lower, upper, step;
```

```
    lower=0;
```

```
    upper=300;
```

```

step=20;

fahr = lower;
while (fahr <= upper) {
    celsius = (5.9/9.0) * (fahr-32.0);
    printf("%3.0f %6.1f\n", fahr, celsius);
    fahr = fahr + step;
}
}

```

Programmanyň görnüşini azajyk üýtgedi. Ol öňki programmadan diňe fahr we celsius üýtgeýän ululyklaryň float tipe degişli edilip beýan edilenligi bilen hem-de öwürmek formulalarynyň has tebigylygy bilen tapawutlanýar. Öňki programmada 5/9 ýazgyny ýazyp bolmaýardy, sebäbi bitinsanlaýyn bölmäniň netijesi nula deň bolardy. Konstantadaky onluk nokat onuň ýüzýän nokatly san hökmünde seredilmegine sebäp bolýar we şeýlelikde 5.0/9.0 ýazgy iki sany ýüzýän nokatly sanyň gatnaşygy ýaly seredilýär we onuň drob böleginiň taşlanmasy bolmaýar. Arifmeiki operasiýa bitin operandlara eýe bolanda ol bitinsanly arifmetikanyň düzgünleri boýunça ýetýär. Eger bir operand ýüzýän nokatly san bolup, beýlekisi bitin san bolsa, onda operasiýa ýerine ýetmezden öň bitin tipli bölegi ýüzýän nokatly tipe öwürüler. Eger biz fahr-32 görnüşde ýazan bolsak, onda 32 san awtomatik usulda ýüzýän nokatly sana öwürüldi. Şeýlede biz ýüzýän nokatly konstantany ýazanymyzda elmydama onluk nokady ulanarys, hatda konstanta bitin tipe degişli bolan ýagdaýynda hem. Munuň özi programmany okaýan adamynyň programmanyň tebigatyna oňat düşünmegine ýardam edýär.

Haýsy ýagdaýlarda bitin ululyklaryň ýüzýän nokatly ululyklara öwürülýänligi 2-nji bapda düýpli seredilýär. Häzir bolsa bir ýagdaýa üns breliň, aşakdaky eýe bolmak

```
fahr = lower;
```

we şerti barlamak

```
while (fahr <= upper)
```

tebigy şekilde işleýär, ýagny operasiýa ýerine ýetmezden öň **int** tipe degişli baha **float** baha öwürülýär.

printf functionynda %3.0f spesifikasiýa ýüzýän nokatly sanyň(häzirki ýagdaýda fahr ululygynyň) giňligi üç pozisiýadan uly bolmadyk meýdançada, onluk nokatsyz we drob böleksiz çapa çykarylýanlygyny aňladýar. %6.1f spesifikasiýa beýlki bir sanyň(celsius) giňligi alty pozisiýaly meýdançada onluk nokatdan soň bir san bilen çapa çykarylýanlygyny aňladýar. Aşakdakylar çapa çykarylar:

```

0      -17.8
20     -6.7
40      4.4

```

Meýdançanyň giňligini we takyklygy görkezmän hem bolar: %6f sanyň alty pozisiýadan köp ýer almajakdygyny aňladýar; %2f – san onluk nokatdan soň iki

bahany alýar, ýöne giňligi çäklendirilmeýär; %f ýöne ýüzyän nokatly sanyň çapa çykarylýanlygyny görkezýär.

%d onluk bitin sany çap etmek

%6d 6 pozisiýaly meýdançanyň içinde onluk sany çap etmek

%f ýüzyän nokatly sany çap etmek

%6f ýüzyän nokatly sany 6 pozisiýaly meýdançada çap etmek

%.2f ýüzyän nokatly sany onluk nokatdan soň 2 sifr bilen çapa çykarmak

%6.2f ýüzyän nokatly sany 6 pozisiýaly meýdançanyň içinde we onluk nokatdan soň 2 sifr bilen çapa çykarmak.

Mundan başgada **printf** functionynda indiki spesifikatorlary ulanmaklyga ýol berilýar: sekizlik san üçin %o, onaltlyk san üçin %x, simwol(litera) üçin %c, string(setir) üçin %s we % simwolyň özi üçin %%.

Ýumuş 1. Tablisanyň yokarsynda sözbaşy çap ediler ýaly edip, temperaturany özgertmek programmasyny kämilleşdirin

Ýumuş 2. Ferengeýtdäki temperaturanyň Selsiýdäki temperatura bilen baglanyşyk tablisasyny çapa çykarýan programma düzüň.

for operatory

Şol bir programmany ýazmaklygyň köp dürli usullary bar. Biziň temperaturany özgertmek programmamyzyň görnüşini üýtgedeliň:

```
#include <stdio.h>
```

```
main()
```

```
{
```

```
    int fahr;
```

```
    for (fahr=0; fahr <= 300; fahr=fahr+20)
```

```
        printf(“%3d %6.1f\n”, fahr, (5.0/9.0)*(fahr-32));
```

```
}
```

Bu programma hem şol bir netijäni çapa çykarýar, ýöne görnüşi gürrüňsiz başga hilidir. Esasy tapawuplanýan ýeri köp üýtgeýän ululyklaryň yoklugyndadyr. Diňe int tipli **fahr** üýtgeýän ululygy bar. Biziň üçin täze bolan **for** instruksiýasynda **fahr** üýtgeýän ululygynyň başlangyç we ahyrky bahalary, şeýle hem onyň uüýtgeýiş ädimi konstanta görnüşinde ýazylypdyr, a Selsiý boýunça temperaturany hasaplaýan aňlatma bolsa aýratyn eýe bolmak instruksiýasy görnüşinde däl-de **printf** functionynyň üçünji argumenti bilen berlipdir.

Soňky üýtgetme aşakdaky umumy düzgüniň ulanylyşyna mysal bolup biler: islendik tipe degişli üýtgeýän ululyga derek şol tipe degişli has çylşyrymly aňlatma hem ulanylyp biliner. Şeýlelikde **printf** functionynyň üçünji argumentiniň ýerine %6.1 spesifikatoryň talabyna görä süýşýän nokatly ululyk bolmaly, şoňa görä-de bu ýerde şol tipe degişli islendik aňlatma bolup biler.

for Instruksiýasy **while** gaýtalanmasynyň umumylaşdyrylan görnüşini beýan edýär. Eger siz ony öň ýazylan while bilen dñeşdirseňiz onda onuň işleýişi aýdyň bolar. Ýaýlaryň içinde biri-birinden nokatlar bilen bölünen üç sany aňlatma bar. Birinji aňlatma – inisalizasiýa

fahr=0

-gaýtalanmaklyga girmezden öň bir gezek ýerine ýetýär. Ikinji – gaýtalanmaklygyň dowam etmeklik şertiniň barlagy

fahr<=300

Şert hasaplanylýar, eger ol **çyn** bolsa onda gaýtalanmaklygyň göwresi(biziň mysalymyzda printf-e bir gezek ýüzlenmek) gaýtalanýar. Ondan soňra indiki ädime geçilýär, bir ädim artdyrylýar:

fahr=fahr+20

we täzedan şert barlanylýar. Haçanda şert **ýalan** bolanda gaýtalanmaklyk tamamlanylýar. while ýaly for gaýtalanmasynyň göwresi bir instruksiýadan ýa-da figuraly ýaýlaryň içine alnan birnäçe instruksiýadan ybarat bolup biler. Bu ýerdäki üç aňlatmanyň(inisializasiýa, şert we ädimiň ulalmagy) ýerinde islendik aňlatma bolup biler.

Programmanyň düşnükli bolmagy nukdaý-nazaryndan seredilip while ýa-da for saýlanylýp alnyp biliner. Haçanda inisializasiýa hem-de ädim artmasy şol bir ululyk arkaly logiki baglanyşykly bolsa we bir instruksiýa arkaly aňladylýan bolsa onda for has amatly bolýar, sebäbi ol while - dan tygşytly ýazylýar, onuň dolandyryjy bölegi bir ýere jemlenendir.

Ýumuş 1. *Temperaturany özgerdýän programmany, tablisany ters tertipde çap eder ýäly(ýagny 300 –den 0 –a çenli) özgerdiň.*

Atlandyrylan konstantalar(hemişelikler)

Temperaturany özgertmek programmasyny tamamlamazdan öň ýene bir pikir aýdalyň. Haçanda programmada 300, 20 ýaly “uly sanlar” dagaşyp ýatan bolsalar bu örän erbet hasaplanylýar. Programmany okajak adam üçin ol sanlaryň hakykatda näme aňladýanlygy barada hiç-hili ýañzytma hem ýok. Mundan başgada olary haýsydyr bir sistematiiki usulda başgalara çalyşmak kyndyr. Beýle sanlar bilen işlemekligi ýeňilleşdirmekligiň bir mümkin bolan usuly olara manyly at dakmakdyr. #define setiri berlen setir üçin *simwiolly ady* ýa-da *atlandyrylan konstantany* kesgitleýär:

#define *at* *goýulýan tekst*

Şu pursatdan başlap *at* ýüze çykan ýerinde(eger ol goşa dyrnagyň içinde ýazylyan setirde we beýleki bir adyň bölegi bolmadyk ýagdaýynda) ol *goýulýan tekst* bilen çalşyrylar. *at* hem üýtgeýän ululyk ýaly görnüşe eýedir: harpdan başlanýan harplaryň we sanlaryň yzygiderligi. *goýulýan tekst* simwollaryň islendik yzygiderligi bolup biler, özüňem olaryň diňe sandan ybarat bolmagy hökman däl.

```
#include <stdio.h>
```

```
#define LOWER 0        /* tablisanyň aşaky çägi */
```

```
#define UPPER 300        /* ýokarky çägi */
```

```
#define STEP 20        /* ädimiň ölçegi */
```

```
/* Ferengeýt we Selsiý boýunça temperatura tablisasyny çap etmek */
```

```
main()
```

```
{
```

```
    int fahr;
```

```

for (fahr = LOWER; fahr <= UPPER; fahr = fahr + STEP)
    printf("%3d %6.1f\n", fahr, (5.0/9.0)*(fahr-32));
}

```

LOWER, UPPER, we STEP ululyklar üýtgeýän ululyklar bolman eýsem atlandyrylan konstantalardyr. Şonuň üçin olar beýan edilmeýärler. Kabul edilen umumy ylalaşyk boýunça, atlandyrylan konstantalaryň atlary kiçi harp bilen ýazylýän adaty üýtgeýän ululyklardan tapawutlylykda, baş harp bilen ýazylýär. #define – setiriň yzynda nokatly oturyň bolmaýanlygyny hem belläp geçeliň.

Literany çapa çykarmak

Indi biz tekstler bilen işlemäge mümkinçilik berýän programmalaryň toplumyna seretmelçi. Siz durmuşda ulanylýan programmalaryň köpüsiniň şu programmalaryň ösdürilen nusgarydygyna göz ýetirersiňiz.

Standart kitaphana giriş-çykyşyň örän ýönekeý modelini goldaýar. Tekst giriş-çykyş nireden gelýänligine ýa-da nirä barýanlygyna bagly bolmazdan literanyň akymy bilen iş salyşýar. *Tekst akymy* – munuň özi her biri nul ýa-da ondan köp litera saklaýan we täze setir literasy bilen tamamlanýan setirlerden ybarat bolan literalaryň yzygiderligidir. Islendik giriş-çykyş akymynyň bu modele gabat gelmekligine göz-gulak bolmak jogapkärçiligi bibliotekanyň üstüne atylandyr: programmist bibliotekany ulananda programmanyň daşynda ol setirleriň nähili görnüşiniň bardygy barada aladalanmaly däldir.

Standart biblioteka bir literany okamak we ýazmak üçin birnäçe functiony özünde saklaýar. Olaryň iň ýönekeýi – **getchar** we **putchar**. getchar functionyna bir gezek ýüzlenilende, tekst akymdan netije görnüşde berilýän *indiki litera* okalýar. Şeýlelikde

```
c = getchar();
```

ýerine ýeteninden soň c üýtgeýän ululyk girizilýän indiki literany kabul edýär. Adatça litera klawiaturadan girizilýär. Faýldan girizilmek 7-nji bapda serediiýär.

Putchar functiony ýüzlenilende ol bir literany çapa çykarýär. Ýagny,

```
putchar();
```

c bitin tipli üýtgeýän ululygyň bahasyny litera görnüşde çapa çykarýar (adatça ekrana). putchar we getchar functionlaryň çagyrylyşy islendik nusgada gezekleşip gelip bilerler. Bu functionlara ýüzlenilişi ýaly tertipde hem çapa çykarmaklyk emele geler.

Faýlyň kopiýasyny almak

getchar we **putchar** funtionlaryň barlygynda, giriş-çykyş barada başga hiç zat bilmäýeniňde-de, örän täsin kän peýdaly programmalary ýazmak bolar. Ýönekeý mysal – bir literadan giriş akymdan çykyş akyma kopiýasyny alýan programma:

```
literany okamak
```

```
while (litera faýlyň soňyny görkezýän litera bolmasa)
```

```
    ýaňky okalan literany çapa çykarmaly
```

```
literany okamak
```

ýokardaky ýazgylary C dilindäki programma görnüşinde beýan edeliň:

```
#include <stdio.h>
/* girişi çikişa kopiýa etmek; 1-nji görnüş */
main()
{
    int c;
    c = getchar();
    while (c != EOF) {
        putchar(c);
        c = getchar();
    }
}
```

Gatnaşyk operasiýasy bolan != “deň däldir” – diýmekligi aňladýar.

Klawiaturadan girizilýän, ekrana çykarylýan ýa-da maşynyň içindäki her bir litera bitleriň birleşmesi görnüşinde kodirlenýär. Char tip, ýörite liter maglumatlary saklamaklyk üçin niýetlenendir, ýöne munuň üçin islendik bitin tipi ulanmak bolar. Ýöne biz düşündirilmegi zerur bolan, möhüm bir sebäbe görä int tipi peýdalanýarys.

Şeýle kynçylyk ýüze çykýar: girişiň soňyny adaty okalýan maglumatlardan nähili tapawutlandyrmaly. Bu meseläniň esasy çözüwi, getchar functiony giriş akymy tamamlananda hakyky literalaryň hiç biri bilen meňzeş bolmadyk bahany gaýdyp bermelidir. Ol baha EOF-a(end of file – faýlyň soňy – diýen sözleriň baş harplary) deňdir. Biziň ulanýan tipimiz, getchar functionynyň berip bilýän hemme bahasyňa “ýeter” ýaly derejede bolmalydyr. Biz bu ýagdaýda char tipini ulanyp bilmeýäris, sebäbi c ululyk char tipiň hemme bahalaryndan başga ýene-de EOF-y saklap bilmelidir. Ine şonuň üçin hem biz char tipini däl-de int tipini ulandyk.

EOF <stdio.h>-da kesgitlenen bitin hemişelikdir. Onuň nähili baha alýanlygy möhüm zat däldir, esasy zat onuň bahasy char tipiň kabul edýän bahasyndan tapawutlanmalydyr. Aýratyn atly atlandyrylan hemişeligiň ulanylmagy, programmanyň takyk san baha bagly bolmazlygyna kepil geçýär, ol ululyklar beýleki C ulgamlardakydan mümkingadar tapawutlanar.

Kopiýasyny alýan programmany has gysga hem ýazyp bolar. C dilinde her bir eýe bolmak, mysal üçin

```
c = getchar()
```

çep gapdaldaky ululyga dakylan, belli bir baha deň bolan aňlatma hökmünde seredilýär. Munuň özi has çylşyrymly aňlatmalaryň içinde hem eýe bolmaklygyň gabat gelip biljekdigini aňladýar. Eger eýe bolmaklygy while gaýtalanmanyň şert barlaýan böleginde ýerleşdirsek, onda faýlyň kopiýasyny alýan programmany aşakdaky ýaly ýazmak bolar:

```
#include <stdio.h>
/* girişi çikişa kopiýa etmek; 2-nji görnüş */
main()
{
```

```

int c;
while ((c = getchar()) != EOF)
    putchar();
}

```

while gaýtalanmasy getchar-yn netijesini c ululyga dakýar we şol wagt barlaýar: ol ululyk “faýlyň soňymy” ýagny EOF-a deňmi. Eger deň däl bolsa, onda while gaýtalanmasy ýerine ýetýär we litera çapa çykarylýar. Giriş akymynyň tamamlanmagy bilen, while öz işini tamamlayar, şonlukda main hem öz işini tamamlayar.

Programmanyň bu görnüşinde getchar-a diňe bir gezek ýüzlenilýär, ýagny girizmek “merkezleşdirilen”. Şonuň netijesinde ol has tygşytly we ýeňil okalýar. Eýe bolmaklygyň şert barlamak bilen gabat gelýän ýagdaýlary size ýygy-ýygydan gabat geler.(Bu ýagdaýa aşa ýykgyň etmeklik, programmany okamaklygy kynlaýar, şonuň üçin bu mümkinçiligi akyly-başly ulanmak gerek).

Şertiň içinde eýe bolmaklygy ýaý şekilli skobkanyň içinde ýazmak zerurdyr. != operasiýanyň möhümligi = operasiýanyň möhümliginden ýokarydyr, şonuň üçin skobkalaryň bolmadyk ýagdaýynda != operasiýa eýe bolmak operasiýasyna çenli ýerine ýeter. Şonuň üçin hem

```
c = getchar() != EOF
```

ýazgy

```
c = (getchar() != EOF)
```

ýazga ekwiwalentdir. A bu bize düýünden gerek zat däl: c üýtgeýän ululyga getchar functionynyň faýlyň soňy alamatyna düşýanlygyna baglylykda 0 ýa-da 1 dakylar.(Bu barada 2-nji bapda has giňişleýin aýdylar).

Ýumuş 1. getchar() != EOF aňlatmanyň 0 ýa-da 1 bahany alýandygyny barlap göz ýetiriň.

Ýumuş 2. EOF-iň bahasyny çapa çykarýan programma düzüň.

Literi sanamak

Aşakdaky programma literi sanamak bilen meşgullanýar; onuň kopiýa alýan programma bilen köp meňzeşlikleri bar.

```

#include <stdio.h>
/* girizilýän literi sanamak; 1-nji görnüs */
main()
{
    long nc;
    nc = 0;
    while (getchar() != EOF)
        ++nc;
    printf(“%ld\n”,nc);
}

```

instruksiýa ++nc, täze ++ operatory görkezýär, ol *bir birlik artdyrmaklygy* aňladýar. Onuň ýerine nc = nc + 1 hem ýazyp bolar, ýöne ++nc has gysga, a köp ýagdaýda effektiv(beýlekä garanda,peýdaly). Edil şonuň ýaly *bir birlik kemeltýän* -- operator hem bar. ++ we - - operatorlar öňünden(++nc) we soňundan(nc++)

gelyän görnüşlerde bolup bilerler. Bu iki görnüş aňlatmalarda dürli baha eýe bolýanlygy, ýöne ++nc we nc++ ikisiniň hem nc ululyga bir birlik goşýalygy 2-nji bapda görkeziler. Häzirki ýagdaýda biz önünden gelyän ýazgynyň üstünde durup geçeriş.

Liter sanaýan programma sanalan literleriň jemini long tipli ululykda saklaýar. Bitin tipli long 32 bitden az däl. Käbir maşynlarda int we long tipleri birmeňzeş ölçege eýe bolsalarda, int tipi 16 bit saklaýan maşynlar hem bar. Olaryň maksimal kabul edip bilýän bahasy 32767, bu beýle bir uly san däl, şonuň üçin hem ol tipe degişli bolan üýtgeýän ululyklaryň dolmagy (переполнение) mümkindir. printf functionynda %ld öwürmek spesifikasiýasy long tipli argumente gabat gelyär.

Eger double(ikeldilen takyklykdaky float tipi) tipi ulanmak bilen has uly aralygy hem almak bolar. Şeýle hem gaýtalanmaklygyň beýleki usulyny görkezmek üçin while derek for ulanalyň

```
#include <stdio.h>
/* girizilýän literi sanamak; 1-nji görnüş */
main()
{
    double nc;
    for (nc=0; getchar() != EOF; ++nc)
        ;
    printf("%.0f\n", nc);
}
```

printf –de %f speaifikator float tipli üýtgeýän ululyklar üçin ulanylýar, ýagny häzirki ýagdaýda double üçin; spesifikator %.0f onluk nokatsyz we drob böleksiz çapa çykarmak(bu mysalda drob san alynmaýar).

Bu ýerde **for** gaýtalanmasynyň göwresi boş, ýagny şert barlamakdan we sanaýjyny artdyrmakdan başga hiç zat etmek gerek däl. Ýöne C diliniň grammatikasynyň düzgüni boýunça for gaýtalanmasynyň göwresi bolmalydyr. Bu talaby, boş instruksiýa diýilip atlandyrylýan izolirlenen nokatly otur ýerine ýetirýär. Biz has düşnükli bolmagy üçin nokatly otury aýratyn setirde goýdyk.

Iň soňunda belläp geçmeli, egerde giriş özünde bir literany hem saklamaýan bolsa, getchar-a birinji gezek ýüzlenilende while ýa-da for gaýtalanmalarynda şert ýerine ýetmez we programmanyň netijesi nul bolar, munuň özi dogry netijedir. Muňa düşünmek möhümdir. while we for gaýtalanmaklarynyň ajaýyp häsiýetleriniň biri şert gaýtalanmaklygyň göwresi ýerine ýetmezden ön barlanylýar. Hiç zat etmeli däl bolsa, hiç zat hem edilmez, hatda şerte baglylykda gaýtalanmaklygyň göwresi bir gezek hem ýerine ýetmän biler. Programma girizilýän literalaryň sany nula deň bolan ýagdaýynda hem özüni dogry(корректно) alyp barmalydyr. while we for gaýtalanmaklarynyň gurluşlary şertiň çäklerinde programmanyň özüni dogry alyp barjaklygyna goşmaça ynam döredýär.

Setiri sanamak

Aşakdaky programma setiri sanaýar. Ýokarda ýatlanylşy ýaly, standart biblioteka her bir setiri täze setir literasy(\n) bilen tamamlanýan setirlerden düzülen tekst akymynyň giriş-çykyş modeli bilen üpjün edýär. Şeýlelikde setiri sanamak täze setir(\n) literasyny sanamaklyga syrygýar.

```
#include <stdio.h>
/* giriş akymyň setirlerini sanamak */
main()
{
    int c,nl;
    nl=0;
    while ((c = getchar()) != EOF)
        if (c == '\n')
            ++nl;
    printf("%d\n", nl);
}
```

Indi gaýtalanmaklygyň göwresini, garamagynda nl sanaýjynyň bahasyny bir birlik ulaltmaklyk ýerleşýän **if** instruksiya emele getirýär. if instruksiya skobkanyň içindäki şerti barlaýar we eger ol çyn bolsa onda onuň yzyndan gelýän instruksiya (ýa-da figuraly skobkanyň içinde ýerleşen instruksiýalaryň toplumy) ýerine ýetýär. Biz ýene-de programmanyň dolandyrylyşynyň düşnükli bolmagy üçin boşluplary ulandyk (haýsy instruksiya haýsy instruksiýany dolandyrýar).

C dilinde ikileýin deňdir alamaty(==) “deňdir” operatoryny aňladýar (Pascaldaky = we Fortrandaky .EQ.). Deňligi barlanylýan operatora Ikileýin deňdir alamatynyň(==) ulanylmagynyň sebäbi ony bir deňdir ulanylýan eýe bolmak operatoryndan tapawutlandyrmak üçindir. Duýdyrýarys: C dilinde programma düzmeklige ýaňy girişenler = ýazmak bilen == göz önünde tutýarlar. Beýle ýagdaýda 2-nji bapda ýazylyşy ýaly, aňlatma görnüşi boýunça ýolbererlikdir, kompilyator hiç-hili duýdyryjy habar bermeýär.

Bir dynagyň içinde ýazylan litera, şol literanyň kodyna deň bolan bitin sany görkezýär(berlen maşynda kabul edilen kodirowka görä). Bu *liter hemişeligi* diýilip atlandyrylýar. Kiçi bitin sanlary ýazmagyň başga bir usuly hem bar. Meselem ‘A’ liter hemişelikdir; ASCII liter ýygynyndysynda onuň bahasy 65-e deňdir, ýagny 65 A harpynyň maşynyň ýadynda bellenilişidir. Elbetde ‘A’ konstanta görnüşinde 65-den göze ýakymlydyr, yagny birinji ýazgynyň manysy has düşnüklidir, ol literalaryň kodirleniş usulyna bagly däl.

String(setir) hemişeliklerinde ulanylýan Esc-yzygiderlikleri liter hemişeliklerinde hem ulanmak bolýar. Ýagny ‘\n’ täze setir literasynyň kody, ol ASCII –de 10 deňdir. ‘\n’ bir litera hökmünde seredilýänligine aýratyn ünsi çekmelidir(onuň kody aňlatmada bitin san görnüşinde seredilýär), “\n” ululyga bolsa setir hemişeligidir. Setir we liter hemişelikleriniň tapawudyna 2-nji bapda düýpli seredilip geçiler.

Ýumuş 1. Boşlugy, tabulýasiýany we täze setiri sanamak üçin programma düzüň.

Ýumuş 2. Girizilýän literalary, yzly-yzyna gelýän boşluklary bir boşluk bilen çalyşmak arkaly çykyş akymyna kopirleýän programma düzmeli.

Ýumuş 3. Girizilýän literalary: tabulýasiýa literasyny \t-a, backspace literasyny \b-a we ters ýapgyt çyzygy \\-a çalşyryp çykyş akymyna kopirleýän programma ýazmaly. Bu hemme tabulýasiýa we backspace literalaryny “görkeklige” mümkinçilik berer.

Söz sanamak

Biziň peýdaly programmalarymyzyň toplumynyň dördünjisi setirleri, sözleri we literalary sanaýan programma bolup, bu ýerde söz diýilip özünde boşluklary, tabulýasiýalary we “täze setir”-leri saklamaýan stringe düşünilýär. Bu programma UNIX sistemasyndaky wc programmanyň sadalaşdyrylan görnüşidir.

```
#include <stdio.h>
#define IN 1 /* sözün içi */
#define OUT 0 /* sözün daşynda */
/* setirleri, sözi we literleri sanamak */
main()
{
    int c, nl, nw, nc, state;
    state = OUT;
    nl = nw = nc = 0;
    while ((c = getchar()) != EOF) {
        ++nc;
        if (c == '\n')
            ++nl;
        if (c == ' ' || c == '\n' || c == '\t')
            state = OUT;
        else if (state == OUT) {
            state = IN;
            ++nw;
        }
    }
    printf("%d %d %d\n", nl, nw, nc);
}
```

Programma her gezek sözün sag literasynyna gabat gelende, sözi sanaýan ululygyň bahasyny bir san artdyrýar. state üýtgeýän ululygy sözün içinde ýa-da daşynda ýerleýändigimizi kesgitleýär. Ilki başda ol ululyga OUT baha dakylýar, ol “sözün daşynda” daşyndadygyny aňladýar. Biz programmanyň has düşnükli bolmagy üçin 1 we 0 bahalaryň özünden dälde IN we OUT atlandyrylan konstantalary ulanmaklygy maslahat berýäris. Biziň bu kiçijik programmamyzda bu usulyň beýän ullakan zady ýok hem bolsa, uly programmalar ýazylanda onuň düşnükliiliginiň ýokarlanmagy, goşmaça sarp edilen kiçijik zähmediň ýerine düşdügi bolar. Siz programmada ulanylýan uly sanlar, atlandyrylan konstantalar görnüşinde ýazylan ýagdaýynda, programma uly üýtgeşmeleriň girizmekligiň ýeňil bolýandygyny görersiňiz.

nl = nw = nc = 0;

setir üç üýtgeýän ululygyň bahasyna nul ýazýar. Bu ýazga üňs berip seretsek ol üýtgeşik bir zat dälir we ulanarlykdyr, ýagny eýe bolmak öžüniň hususy bahasy bolan aňlatmadyr we eýe bolmak operasiýasy sagdan çepe tarap ýerine ýetýär. Ol setir aşakdaky setire ekwiwalentdir:

nl = (nw = (nc = 0));

|| - operatory ÝADA atly logiki operasiýany aňladýar, şeýlelikde aşakdaky setir:

if (c == ' ' || c == '\n' || c == '\t')

“eger c ululyk boşluk ýa-da “täze setir” ýa-da tabulýasiýa bolsa” – diýilip okalýar. (\t esc-zygiderlik tabulýasiýa literasyny aňladýandygyny yatlalyň.) Şeýle hem logiki WE – ni aňladýan && operator bar. Onuň möhümligi || operatoryňkydan ýokarydyr. && ýada || operatorlar bilen bagly aňlatmalar çepden saga hasaplanýar, özünem şertiň çynlygy ýada ýalanlygy belli bolan halatynda hasaplamanyň tagtadyljakdygyna güwä geçilýär. Eger c ululyk boşluk bolsa onda c ululygyň “täze setir”-e ýada tabulýasiýa deňdigini barlamak gerek däl. Bökler ýagdaýlarda beýle usulda hasaplamak beýle bir möhüm däl, ýöne has çylşyrymly ýagdaýlarda ony ulanmaklygyň manysy bardyr, biz onyň bilen gelejekde iş salşarys.

Ýokarky mysalda *else* sözi hem gabat geldi, ol *if* –däki şert ýerine ýetmedik ýagdaýynda ýerine ýetirmeli instruksiýany görkezýär. Umumy görnüşde şertli instruksiýa şeýle ýazylýar:

if (aňlatma)

instruksiýa1

else

instruksiýa2

if-else konstruksiýasynda görkezilen instruksiýalaryň diňe biri ýerine ýetýär. Eger *aňlatma* çyn bolsa, onda *instruksiýa1*, eger ýalan bolsa onda *instruksiýa2* ýerine ýetýär. Bu instruksiýalaryň her biri ýaha bir instruksiýa, ýa-da figuraly skobkalaryň içinde ýazylan birnäçe instruksiýalardan ybaratdyr. Biziň programmamyzda else-den soň iki sany şonuň ýaly instruksiýany dolandyryýan if instruksiýa bar.

Ýumuş 1. Söz sanaýan programmany nähili testirlemeli? Nähili ýalňyşlygy bu programmanyň duýmazlygy mümkin.

Ýumuş 2. Öziniň girişindäki sözleriň her birini aýratyn setirde çapa çykaryýan programma düzüň.

Massiwler

A indi bolsa aýratynlykda her-bir sifri, boşluk literalaryny (boşluk, tabulýasiýa we täze setir) we beýleki literalary sanaýan programma ýazalyň. Bu programmanyň ulanyljak ýeri az hem bolsa, ol bize bir mysalda C diliniň ýene-de birnäçe mümkinçiliklerini görkezýär. Girizilýä literalaryň on iki kategoriýasy bar. On sany sifri sanaýjylary on sany aýratyn ululyklarda saklaman, bir massiwde ýerleşdirmek amatly. Ol programmanyň bir görnüşi:

#include <stdio.h>

```

/* sifrleri, boşluk we beýleki literalary sanamak */
main()
{
    int c, i, nwhite, nother;
    int ndigit[10];
    nwhite = nother = 0;
    for (i=0; i < 10; ++i)
        ndigit[i] = 0;
    while ((c = getchar()) != EOF)
        if (c >= '0' && c <= '9')
            ++ndigit[c-'0'];
    else if (c == ' ' || c == '\n' || c == '\t')
        ++nwhite;
    else
        ++nother;
    printf("sanlar=");
    for (i = 0; i < 10; ++i)
        printf(" %d", ndigit[i]);
    printf(", boşluklar = %d, beýlekiler = %d\n", nwhite, nother);
}

```

Eger bu programmany işe goýbersek hem-de programmanyň tekstini giriş hökmünde girizsek, onda aşakdaky netije alnar:

sanlar = 9 3 0 0 0 0 0 0 1, boşluklar = 141, beýlekiler = 347

```
int ndigit[10]
```

deklarasiýa 10 bahadan ybarat, int tipli massiwi kesgitleýär. C dilinde massiwiň elementleri elmydama nuldandan başlap belgilenýär, şeýlelikde bu massiwiň elementleri ndigit[0], ndigit[1], ..., ndigit[9] görnüşde bolar, bu ýagdaýy for – gaýtalanmada (inisializasiýada we massiw çapa çykarylanda) göz önünde tutulmalydyr.

Bitin üýtgeýän ululyklardan (mysal üçin i) we bitin konstantalardan emele gelen aňlatmalar indeks bolup hyzmat edip biler.

Bu programma sifrleriň kodirowkasynyň kesgitli häsiýetine esaslanýar. Mysal üçin

```
if (c >= '0' && c <= '9') . . .
```

barlag, c üýtgeýän ululykda sifriň saklaýanlygyny kesgitleýär. Eger bu şeýle bolsa, onda

c – '0'

aňlatma sifriň san bahasy bolmaly. Bu aýdylanlar diňe '0', '1', . . . , '9' bahalaryň hatarynda her bir indiki baha öňkiden bir san uly bolan halatynda dogrudyr. Literleriň hemme naborynda bu düzgün saklanylýar.

Kesgitleme boýunça char tipiň bahalary kiçi bitin sanlardyr, ýagny char tipli üýtgeýän ululyklar we hemişelikler arifmetiki aňlatmalarda int tipli bahalara meňzeşdir. Bu tebigydyr we amatlydyr; mysal üçin c-'0' aňlatma c ululykdaky 0 – dan 9-a çenli bitin bahany kabul edip bilýär. Şeýlelikde bu aňlatmanyň bahasy ndigit massiwiň dogry indeksi bolup hyzmat edýär.

Programmanyň aşakdaky bölegi literanyň sanlygyny, boşluk literasydygyny ýa-da başga bir literadygyny kesgitleýär.

```
if (c >= '0' && c <= '9')
    ++ndigit[c-'0'];
else if (c == ' ' || c == '\n' || c == '\t')
    ++nwhite;
else
    ++nother
```

Aşakdaky konstruksiýa

```
if (şert1)
    instruksiýa1
else if (şert2)
    instruksiýa2
...
...
else
    instruksiýan
```

programmada bar bolan birnäçe alternatiw ýoldan birini saýlamak üçin ýygýgydan peýdalanylýar. *Şertleriň* haýsy hem bolsa biri ýerine ýetýänçä, ýokardan aşak hasaplanylýar; şert ýerine ýetende oňa degişli instruksiýa hem ýerine ýetirilýär we бүтін konstruksiýa öz işini tamamlýar. (Islendik instruksiýa figuraly skobkalaryň içinde ýazylan instruksiýalaryň toplumy bolup biler) Eger şertleriň hiç birisi ýerine ýetmeýän bolsa, onda else sözünüň yzynda ýerleşen iň soňky instruksiýa ýerine ýetýär, (eger şeýle instruksiýa bar bolsa). Eger iň soňky else we instruksiýa ýok bolsa, onda düýbünden hiç zat ýerine ýetmeýär.

Ilkinji if bilen iň soňky else sözünüň arasynda islendikçe mukdarda aşakdaky kombinasiýa bolup biler

```
else if (şert)
    instruksiýa
```

Haçanda olar birnäçe bolsa, onda olary biziň bu ýerde görkezişimiz ýaly formatirlemek amatly bolýar. Eger her bir indiki if sözünü öňündäki else sözüne görä saga süýşirip ýazsak, onda barlaglaryň juda köp bolan halatynda, tekstiň sahypanyň sag tarapky çäğine çeňli baryp ýetmegi mümkin.

3-nji bapda gürrüni ediljek switch instruksiýasy C dilinde köpýolly şahalanmanyň gurnamaklygyny başga bir usuly bilen üpjün edýär. Bu usul haçanda geçmek şerti, bitin tipli aňlatmanyň bahasy konstantalaryň berlen toplumynyň birine gabat elen halatynda ulanmak has amatly bolýar. 3.4 bölümde bu programmanyň switch operatoryny ulanmak bilen yazylan görnüşi getirilýär.

Ýumuş 1. Girizilen sözleriň uzynlyklarynyň gistigrammasyny çapa çykarýan programma düzüň. Gistogrammany kese çyzyklaryň kömegi bilen çekmek ýeňildir. Gistogrammany dik çyzyklaryň kömegi bilen gurmak – has kyn mesele.

Ýumuş 2. Girizilen literalaryň gabat geliş ýygylgyny boýunça gistogrammany çapa çykarýan programma düzüň.

Funksiýalar

C dilinde function Fortran dilindäki kömekçi programma we function, Pascal dilindäki procedure we function ýalydyr. Function käbir hasaplama aýratyn ýazmaklyga we onuň gurluşy barada hiç zat bilmezden ony ulanmaklyga mümkinçilik berýär. Function ýazylandan soň, olaryň gurluşy barada hiç zat bilmekligiň geregi ýok, diňe olaryň näme edýänligini bilmek ýeterlikdir. C dilinde functionlary ulanmaklyk amatly, aňsat we peýdalydyr (effektiw). Kähalatlarda siz diňe bir gezek çagyrylýan, gysgajyk functionlara gabat gelersiňiz; olar diňe bir maksat üçin – programmanyň düşnükli bolmagy üçin ýazylandyr.

Biz şu çaka çenli printf, getchar we putchar ýaly taýýar functionlardan peýdalandyk, indi öz hususy birnäçe functionlarymyzy ýazmaklyga wagt boldy. C dilinde Fortran dilindäki ýaly ** derejä götermek operatory ýok. Şonuň üçin bitin m sany bitin položitel n derejä göterýän power(m,n) funkstionynyň mysalynda functionlary kesgitlemeklige seredeliň. power(2,5) functionyň bahasy 32-ä deň. Hakykatda bu functionyň praktiki amiyeti az, ýagny ol diňe kiçi bitin derejeler bilen işläp bilýär, ýöne ol functionyň işleýşine düşünmek üçin oňat mysaldyr. (Standart bibliotekada x^y –ni hasaplaýan pow(x,y) bar.)

Şeýlelikde biziň power we onuň hyzmatyndan peýdalanýan baş functionymyz bar, tutuş programmamyz aşakdaky görnüşde bolar:

```
#include <stdio.h>
int power(int m, int n);
/* power functionynyň testi */
main()
{
    int i;

    for (i = 0; i < 10; ++i)
        printf("%d %d %d\n", i, power(2,i), power(-3,i));
    return 0;
}

/* base –i n-nji derejä göterýär; n >= 0 */
int power(int base, int n)
{
    int i, p;

    p = 1;
    for (i = 1; i <= n; ++i)
```

```

        p = p * base;
    return p;
}

```

Islendik functiony kesgitlemekligiň aşakdaky görnüşi bar:
netijäň tipi functionyň tipi(parametrleriň toplumy, eger bar bolsa)

```

{
    deklarasiýa
    instruksiýa
}

```

functionlaryň kesgitlenişi islendik tertipde, bir ýa-da birnäçe faýlda (source) bolup biler, ýöne islendik function tutuşlygyna bir faýlda ýerleşmelidir. Eger programmanyň teksti birnäçe faýla bölünen bolsa, onda ony kompilirlmek hem-de ýerine ýetirmek üçin bir faýlda ýerleşdirilen programma garanda köp zatlary ýazmaly bolarys; ýöne bu aýdylanlar, eýýäm programmirlme diline dälde, operasion ulgama degişlidir. Häzirikçe biz bu functionlar bir faýlda ýerleşen diýip hasaplaýarys, şeýlelikde siziň C dilinde ýazylan programmany işletmek üçin öwrenen zatlaryňyz bu ýerde ýeterlik bolar.

Aşakdaky setirde main functionyndan power functionyna iki gezek ýüzlenilýär.

```
printf(“%d %d %d\n”, i, power(2,i), power(-3,i));
```

power functiony her gezek çagyrylanda oňa iki argument berilýär we her gezek baş programma jogap hökmünde bir bitin sany kabul edýär, soňra ol san degişli format boýunça çap edilýär. power(2,i), aňlatmanyň bölegi hökmünde 2 ýa-da i ýaly bitin sany özünde saklaýar. (Hemme functionlaryň netijesi bitin san bolmaýar, bu barada soň, 4-nji bapda aýdylar)

power functionyň birinji setirinde şeýle kesgitleme bar:

```
int power(int base, int n)
```

netijäniň we functionyň hem-de onuň parametriniň tipini görkezýär. Parametrleriň ady powerde lokallaşdyrylandyr, munuň özi islendik beýleki function üçin gizlidir, şonuň üçin beýleki functionlaryň öz maksatlary üçin şeýle atlary ulanmaklyga erkinlik bilýärler. Şol aýdylanlar i,p üýtgeýän ululyklar üçin hem dogrudyr: power functionyndaky i bilen main functionyndaky i ululyklaryň arasynda hiç hili umumylyk ýokdyr.

Biz indi function kesgitlenende, skobkalaryň içinde ýerleşdirilen üýtgeýän ululyklara parametr, functiona ýüzlenilende ulanylýan bahalara bolsa argument diýip atlandyrjakdyys. Biz kä halatda şol bir manyda *formal argument* we *fakt argument* diýen düşüňjeleri ulanjakdyrys.

power functionynyň hasaplamasynyň netijesi, return instruksiýasynyň kömegi bilen main functionyna gaýtarylýar. return sözüniň yzynda islendik aňlatma bolup biler:

```
return aňlatma;
```

Functionyň netije bermegi hökman däl. Eger return instruksiýasynyň aňlatmasy bolmasa, onda ol hiç hili baha gaýtarmazdan, özüni çagyran functiona dolandyrmaklygy geçirýär. Eger hasaplama netijesinde biz functionyň ýapylýan

figura skobkaly soňuna ýetsek hem edil ýokardaky ýaly, hiç hili baha gaýtarmazdan, özüni çagyran functiona dolandyrmaklygy geçirýär. Ýüz tutýan functionyň alynýan netijeden ýüz öwürýän ýagdaýlary hem bolup biler.

Siz ähtimal main functionynyň soňundaky return instruksiýasyna üns berensiňiz. Umuman aýdanymyzda main hem beýleki functionlar ýaly function bolup ol öz gezeginde özüne edilen çakyyga jogap bermelidir. Baş function bolan main functiony özüni işe goýberen operasion ulgama netije gaýtarýar. Adatça netije nul bolup, ol işiň normal gutaranlygy barada habar berýär. Eger netije nula deň bolmasa, ol programmanyň adaty däl ýa-da ýalňyş tamamlananlygyny aňladýar. Biz şu çaka çenli programmanyň has düşnükli bolmagy üçin, main functionynda returny ýazmadyk, ýöne indi biz ony ýazarys, sebäbi programma öz tamamlayyş netijesi barada operasion ulgama habar bermelidir.

```
int power(int m, int n);
```

deklarasiýa main functionyň önünde durmak bilen ol power functionynyň iki sany bitin argumente garaşýanlygyny we bitin tipli netijäni gaýtarýanlygyny aňladýar. *Functionyň prototipi* diýilip atlandyrylýan bu beýan etme power functionynyň hemme çagyryşlary we kesgitlenmesi bilen baglanyşykly bolmaly. Eger functionyň kesgitlenişi ýa-da çagyrylyşy özüniň prototipine gabat gelmeýän bolsa ol ýalňyşlykdyr.

Parametrleriň atlary ylalaşygy talap etmeýär. Umuman olar prototipde islendik bolup biler ýa-da düýbünden bolman hem biler, ýagny prototip aşakdaky ýaly hem ýazylyp biliner:

```
int power(int, int);
```

Ýöne öýlanylyp goýulan atlar programmanyň düşnüpli bolmagyny üpjün edýär, şonuň üçin biz köplenç şeýle atlary ulanarys.

Taryhy bellik. ANSI-C diliniň öňki wersiýalardan esasy tapawudy hem functionyň beýan edilişinde we kesgitlenilişindedir. C diliniň ilkinji wersiýasynda power functiony aşakdaky görnüşde yazmak talap edilýär:

```
/* power base ululygy n-nji derejä götermek; n >= 0 */  
/* (C diliniň köne stilindäki görnüşi) */  
power (base, n)  
int base, n;  
{  
    int i, p;  
    p = 1;  
    for(i = 1; i <= n; ++i)  
        p = p * base;  
    return p;  
}
```

Bu ýerde parametrleriň atlary ýaý şekilli skobkalarda görkezilýär, a olaryň tipleri bolsa ilkinji açylýan figuraly skobkanyň önünden berlipdir. Parametriň tipi baradaky görkezme bolmadyk ýagdaýynda, ol int tipine degişli hasap edilýär.(Functionyň göwresi üýtgemä sezewar bolmandyr.)

power functionynyň programmynyň başyndaky beýan edilişi C diliniň ilkinji wersiýasyna laýyklykda aşakdaky görnüşde bolamlydyr:

```
int power();
```

Parametrleriň toplumyny görkezmek bolmaýar, şonuň üçin hem kompilýatoryň power functionyna ýüzlenmekligiň dogrulygyny barlamak mümkinçiligi ýok. Ýagny power functionynyň beýan edilişi bolmadyk ýagdaýynda function bitin tipli netijäni berýär diýilip düşünilýär, onda bu ýagdaýda beýan etmekligi doly aýyrmak hem bolar.

Functionlaryň prototipleri üçin täze barlag(sintaksis) argumentleriň sany we olaryň tipi barada ýalňyşlygy kesgitlemekligi ýeňilleşdirýär. Functiony beýan etmekligiň we kesgitlemekligiň köne barlagy ANSI standarty tarapyndak rugsat berilýär, ýöne siziň kompilýatoryňyz täze sintaksisi(barlagy) goldaýan bolsa biz diňe şony ulanmaklygy örän maslahat berýäris.

Ýumuş 1. Temperaturany özgertermekligiň özüni özbaşdak function görnüşinde bölmek arkaly temperaturany özgerdýän programmany täzedan ýazyň.

Argumentler. Bahasy boýunça çagyrmak

C dilindäki functionlaryň bir häsiýeti, beýleki dillerde programma düzmek dillerini ulanýanlar üçin, mysal üçin Fortran dilini ulanýanlar üçin täzelik bolar. C dilinde functionyň hemme argumentleri “bahasy boýunça” geçirilýär. Munuň özi, çagyrylýan functiona argumentleriň özi dälde, argumentleriň bahalary wagtlaýyn ululyklarda geçirilýär. Argumentleri geçirmekligiň beýle usuly argumentleriň lokal kopiýalary bilen dälde, gönüden-göni özi bilen işleýän, Fortrandaky “yuz tutmak boýunça çagyrmak”-dan we Pascal-daky parametrleriň var spesifikasiýasyndan birneme tapawutlanýar.

C diliniň kömekçi programmalary ulanmakdaky esasy tapawudy, çagyrylýan function çagyrylan functionyň üýtgeýän ululygyny(argumentini) üýtgedip bilmeýär: ol diňe onuň bölek, wagtlaýyn kopiýasyny üýtgedip bilýär.

Özünem bahasy boýunça çagyrmaklygy diliň kem tarapy dälde, artykmaç tarapy hasaplamaly. Diliň bu häsiýetine laýyklykda, “keseki” üýtgeýän ululyklarynyň sany az bolan, ýagny parametrlere çagyrylýan kömekçi programmalaryň lokal inisializirlenen ululyklary hökmünde seretmek bilen has tygşytly programma ýazmak bolýar. Bu aýdylanlara mysal edip power functionynyň ýene bir görnüşine seredeliň.

```
/* power base ululygy n-nji derejä götermek; n >= 0; 2-nji görnüş */
```

```
int power(int base, int n)
{
    int p;
    for(p = 1; n > 0; --n)
        p = p * base;
    return p;
}
```

Bu ýerde n parametr wagtlaýyn ululyk bolup, ol ululygyň bahasy tä nul bolýança for gaýtalanmasynda kemeldilmek arkaly sanamak amala aşyrylýär. Bu ýagdaýda

gaýtalanmany sanamak üçin goşmaça i ululygyň zerurlygy ýok. Biz power-iň içinde n ululygyň bahasyny üýtgetsek hem ol argumentiň özüne hiç-hili täsir etmez, sebäbi ol function argumentiň kopiýasy bilen çagyrylýar.

Zerurlyk bolan ýagdaýynda çagyrylýan functionyň özüni çagyryan programmanyň üýtgeýän ululygyny üýgedip biler etmek mümkinçiligi hem bar. Onuň üçin çagyryan programma üýtgedilmäge degişli ululyklaryň adreslerini (salgylaryny, üýtgeýän ululyklaryň görkezijilerini) degişli parametrleri görkeziji hökmünde ulanmak arkaly çagyrylýan functiona geçirmelidir. Görkezijilere degişli zatlaryň hemmesine biz 5-nji bapda seredip geçäris.

Massiwi argument görnüşinde geçirmekligiň mehanizmi bolsa başgaçadyr. Haçanda massiwiň ady functionda argument hökmünde ulanylan ýagdaýynda, onda functiona massiwiň başlangyç adresini görkezýän ululyk geçirilýär; massiwiň hiç-hili elementleri kopirlenmeýär. Function kabul edip alan ululygyny Indeksirlemegiň kömegi bilen massiwiň islendik elementi bilen işläp bilýär. Bu barada indiki bölümde gürrüň ederis.

Literleriň massiwi

C dilinde massiwleriň iň bir ýaýran görnüşi – literleriň massiwidir. Liter massiwleriň we olar bilen işleýän functionlaryň ulanylyşyny görkezmek maksady bilen, tekst setirleriniň toplumyny okaýan hem-de olaryň içinden iň uzynyny çapa çykarýan programma ýazalyň.

Onuň işleýşi diýseň ýönekey:

```
while (ýene-de setir barmy?)
    if (bu setir öňkileriň hemmesinden uzyn)
        ony ýatda saklamaly
        onuň uzynlygyny ýatda saklamaly
    iň uzyn setiri çapa çykarmaly
```

Onuň işleýişinden görnüşi ýaly, programma tebigy görnüşde bölekler bölünýär. Olariň biri täze setiri girizýär, beýlekisi ol setiri barlaýar, üçünji ony ýatda saklaýar, a galanlary bolsa hasaplaýyş prosessini dolandyryar.

Görnüşü ýaly bu iş birnäçe takyk bölekler bölünýär, ol bölekleri şonlugyna hem C diline geçirsek programma düşnükli bolar. Şonuň üçin ilki bilen setiri okamak üçin aýratyn **getline** functionyny ýazalyň. Biz bu functiony beýleki maksatlar üçin hem peýdaly bolar ýaly ýazmaklyga çalyşarys. getline functiony ilki bilen faýlyň(girişin) soňy barada habar bermelidir, has gowysy ol setiriň uzynlygyny gaýdyp berse (faýlyň uzynlygy nul, haçanda faýlyň soňuna ýetilende). Nul faýlyň soňuny görkezýän alamat bolup hyzmat edip biler, sebäbi uzynlygy nula deň bolan faýl ýokdyr, hatda diňe bir “täze setir” literasyny saklaýan faýlyň hem uzynlygy 1-e deňdir.

Haçanda biz öňki setirleriň içindäki iň uzyn setirden hem uzyn setire gabat gelsek ony bir ýerlerde ýatda saklamaly bolarys. Bu ýerde şol tapylan setiri ähtibar bir ýere ýerleşdirmek üçin **copy** atly ikinji functionyny ulanarys.

Iň soňunda **getline** we **copy** functionlaryny dolandyrmak üçin baş programma gerek. Biziň programmamyz tutuşlaýyn görnüşde aşakdaky bolar:


```

#include <stdio.h>
#define MAXLINE 1000 /* girizilen setiriň maksimal ölçegi */
int getline(char line[], int maxline);
void copy(char to[], char from[]);
/* iň uzyn setiri çapa çykarmak */
main()
{
    int len; /* alnan setiriň uzynlygy */
    int max; /* barlanylan setirleriň içinden iň uzyny */
    char longest[MAXLINE]; /* alynan setir */
    char longest[MAXLINE]; /* iň uzyn setir */
    max = 0;
    while ((len = getline(line, MAXLINE)) > 0)
        if (len > max) {
            max = len;
            copy(longest, line);
        }
    if (max > 0) /* iň bolmanda bir setir barmy ? */
        printf("%s", longest);
    return 0;
}

/* getline: s setiri okaýar, onuň uzynlygyny gaýtarýar */
int getline(char s[], int lim)
{
    int c, i;
    for (i=0; i<lim-1 && (c=getchar())!=EOF && c!='\n'; ++i)
        s[i] = c;
    if (c == '\n') {
        s[i] = c;
        ++i;
    }
    s[i] = '\0';
    return i;
}

/* copy: 'from'-dan 'to'-a kopirleýär; to islendik uly bolup biler */
void copy(char to[], char from[])
{
    int i;
    i = 0;
    while ((to[i] = from[i]) != '\n')
        ++i;

```

}

Biz programmanyň başynda beýan edilen **getline** we **copy** functionlary, main bilen bir faýlda ýerleşýär diýip hasaplaýarys.

main we getline functionlary iki sany argument hem-de yzyna gaýtarylýan netije boýunça täsir edişýärler. getline – de argumentler aşakdaky setir bilen kesgitlenýär

int getline(char s[], int lim)

Görşümiz ýaly onuň birinji argumenti s massiw, a ikinjisi lim bolsa bitin tipli ululyk. Kesgitlenende massiwiň ölçegini görkezmek, ätiýaçdan zerur bolan ýady eýelemek bilen baglanyşyklydyr. getline-yň özünde s massiwiň uzynlygyny görkezmekligiň zerurlygy ýok, sebäbi onuň uzynlygy main-da görlezişlidir. getline özüne ýüz tutan functiona netijäni gaýtarmak üçin, edil power functionynda edilişi ýaly, return-y ulanýar. Ýokarky setirden görnüşi ýaly getline int tipli bahany yzyna gaýtaryar, egerde functionyň tipi görkezilmedik bolsa onda ol int tipli diýilip hasaplanylýar, şeýlelikde getline functionyň önünden int sözüni aýyrsak hem bolar.

Käbir functionlar netije berýärler, a käbirleri bolsa hiç-hili netije bermän diňe käbir täsirleri ýerine ýetirýär(copy ýaly). copy functionynda netijäniň tipiniň ýerinde void sözi dur. Bu söz functionyň hic-hili netije bermeýänligini aýdyň görkezýär.

getline functiony döredýän massiwiniň soňunda, onuň soňuny görkezmek maksady bilen '\0' literany ýerleşdirýär. Bu ylalaşyk

“hello\n”

tipli literlerden ybarat setiriň soňunda hem saklanýar. Häzirki ýagdaýda soňunda '\0' liter ýerleşdirilen, literlerden düzülen massiw emele getirilýär.

h	e	l	l	o	\n	\0
---	---	---	---	---	----	----

printf – iň formatynda %s spesifikasiýa, oňa degişli argument ýokardaky ýaly görnüşdäki string(setir) bolmalydyr. copy functiony hem öz işleýişinde okaýan argumenti '\0' litera bilen gutarýar hasaplaýar, özünem beýlekiler bilen bir hatarda ol literany hem kopirleýär.(Hemme aýdylanlar adaty tekstiň içinde '\0' litera düş gelmeýär hasaplaýar.)

Hatda şular ýaly kiçijek programmada hem käbir kynçylyklaryň ýüze çykýanlygyny belläo geçmek gerek. Mysal üçin, eger ýol bererlikden uzyn setir gabat gelse main näme iş etmeli? getline functiony ähtibar işleýär: eger massiw dolsa, '\0' literasy gabat gelmedik halatynda hem geçirmekligi tamamlar. Baş programma getline functiondan girizilen setiriň uzynlygyny alanyndan sön, onuň uzynlygy MAXLINE bilen gabat gelýän bolsa, onda bu aýratyn ýagdaýy gözöňünde tutmak üçin programmada käbir zatlary goşmak bolar. Programmanyň gysga bolmagynyň hatyrasyna biz oňa bu ýerde seretmeýäris.

getline functionynyň ulanyjylary girizilýän setiriň nä derejede uzyn boljaklygyny önünden bilmeýärler, sonuň üçin getline çogmaklyk(overflow) şertini barlaýar. copy functionynyň ulanyjylary üçin kopirlenýän setiriň uzynlygy

belli(ýa-da olar bu barada bilip bilerler), şonuň üçin hem bu ýerde goşmaça barlagyň geregi ýok.

Ýumuş 1. Ýokardaky programmanyň main-yny ol islendik uzynlykdaky iň uzyn setiri, onuň ölçegine hiç-bir çaklendirmesiz çapa çykarar ýaly edip täzedden ýazyň.

Ýumuş 2. Girizilýän setirleriň içinden uzynlygy boýunça 80 literadan uzynlaryny çapa çykarmak üçin programma düzüň.

Ýumuş 3. Hebir girizilýän setirde yzly-yzyna gelýän probel we tabulýasiýa literalaryny bir probele çalyşýan hem-de boş setireri aýyrýan programma düzüň.

Ýumuş 4. s stringdäki literalary ters tertipde ýerleşdirýän reverse(s) functionyny düzüň. Her bir girizilýän setiri tersine öwürýän programma düzüň, ol programmada bolsa reverse(s) functionyny ulanyň.

Daşky üýtgeýän ululyklar we olaryň täsir edýän çägi

line, longest üýtgeýän ululyklary diňe main functionyna degişli, ýa-da aýdylyşy ýaly şol fubctionyň içinde lokallaşdyrylan. Olar main-iň içinde kesgitlenen bolansoň, başga hiç bir function göniüden-göni olara ýüzlenip bilmeýär. Beýleki functionlaryň üýtgeýän ululyklary barada hem şony aýtmak bolar; mysal üçin getline-daky i-iň, copy-daky i bilen hiç-hili baglanyşygy ýokdur. Functionyň her bir lokal üýtgeýän ululygy, şol functiona ýüz tutulan pursatynda döreýär we functiondan çykylanda bolsa olar ýitip gidýärler. Olar ýaly üýtgeýän ululyklara, beýleki dilleriň terminologiýasynda aýdylyşy ýaly *awtomatik* üýtgeýän ululyklar diýilýär.(4-nji bapda, lokal ululyklara çagyryşlaryň arasynda hem öz bahasyny saklap bilmek mümkinçiligini döredýän static diýilip atlandyrylýan ýadyň klasyna seredilip geçilýär.)

Şeýlelikde awtomatiki ululyklar functiona girilmegi we ondan çykylymagy bilen bir wagtda emele gelýär we ýitip gidýär, olar öz bahalaryny ol çakylykdan beýleki bir çakylyga çenli saklamaýarlar, şonuň üçin functiona täzedden ýüzlenilende olaryň bahalary täzedden görkezilmelidir. Eger şeýle edilmese onda olar özünde “musor” saklarlar.

Awtomatik ululyklaryň alternatiwasy görnüşinde, atlary boýunça islendik functiondan ýüzlenmeklige mümkinçilik berýän *daşky* ululyklary mysal getirmek bolar.(Bu mehanizm Fortrandaky ýadyň comman bölegine we Pascaldaky iň daşky blokda beýan edilen ululyklar bilen meňzeşdir.) Şeýlelikde, *daşky* ululyklara hemme functionlardan ýüzlenip bolýanlygy üçin, olary functionlary maglumat boýunça baglanyşdyrýan argumentleriň ornunda ulanmak bolar. Bulardan başgada daşky ululyklar, functionlar aktiwleşende ýüze çykman, hemişe barlygy üçin olar function öz işini tamamlandan soň hem şol ýerde kabul eden bahalaryny özlerinde saklaýarlar.

Daşky ululyklar hemme functionlaryň daşynda, diňe bir gezek kesgitlenýärler; şonuň üçin olara özbaşdak ýat bölünip berilýär. Daşky ululyklar olary ulanmakçy bolýan hemme functionlarda beýan edilmelidir(deklerasiýa). Bu beýan edilmegi ululygyň tipi barada maglumat saklamalydyr. Beýan etmeklik açyk extern instruksiýasy görnüşde ýa-da gerek maglumat kontekstden alynýan açyk däl görnüşde bolup biler. Aýdylanlary has takyklaşdyrmak üçin, iň uzyn setiri çapa

çykarýan programmany line, longest we max ululyklary daşky ululyk görnüşinde beýan etmek arkaly gaýtadan ýazalyň. Bu üç functionlaryň hemmesiniň çagyryşynda, beýan edilişinde we göwresinde üýtgeşmeleriň girizilmegini talap edýär.

```
#include <stdio.h>
#define MAXLINE 1000 /* girizilýän setiriň max. uzynlygy */
int max; /* seredilen setirleriň içinden iň uzyny */
char line[MAXLINE]; /* häzirki setir */
char longest[MAXLINE] /* iň uzyn setir */
int getline(void);
void copy(void);
/* n uzyn setiri çap etmek */
main()
{
    int len;
    extern int max;
    extern char longest[];
    max = 0;
    while((len = getline()) > 0)
        if (len > max) {
            max = len;
            copy();
        }
    if (max > 0) /* iň bolmanda bir setir */
        printf("%s", longest);
    return 0;
}

/* getline: ýözüteleşdirilrň görnüşi */
int getline(void)
{
    int c, i;
    extern char line[];
    for(i = 0; i < MAXLINE-1
        && (c = getchar()) != EOF && c != '\n'; ++i)
        line[i] = c;
    if (c == '\n') {
        line[i] = c;
        ++i;
    }
    line[i] = '\0';
    return i;
}
```

```

/* copy: ýörüteleşdirilen görnüşi */
void copy(void)
{
    int i;
    extern char line[], longest[];
    i = 0;
    while ((longest[i] = line[i]) != '\0')
        ++i;
}

```

main, getline we copy üçin daşky ululyklar biziň mysalymyzyň başynda kesgitlenýär, olaryň tipi kesgitlenýär we ýer bölünip berilýär. Daşky ululyklary kesgitlemek, lokal ululyklary kesgitlemekden tapawutlanmaýar, diňe olaryň functionlaryň çäginde daşarda ýatanlygy üçin olara daşky ululyklar hasaplanylýar. Functionyň daşky ululygy ulanyp bilmegi üçin, ilki bilen oňa ol ululyk barada habar bermeli. Bu bolsa *extern* sözüniň kömegi bilen beýan edilýär, bu beýan etmeklik daşky ululygy beýan edilişinden diňe *extern* söziden başlanýanlygy bilen tapawutlanýar.

Käbir ýagdaýlarda *extern* beýan etmesini ýazman hem bolar. Eger programma ýazylyan faýlda daşky ululygyň kesgitlenişi ony ulanýan functiondan ýokarda bolsa, onda *extern* beýan etmesiniň zerurlygy ýok. Şeýlelikde main, getline we copy functionlarynda *extern* beýan edilmesiniň geregi ýok. Adatça daşky beýan etmeklik faýlyň başynda ýerleşýär, şonuň üçin hem hemme functionlar üçin *extern* beýan etmeleriniň geregi bolmaýar.

Eger programma birnäçe faýlda ýerleşip, daşky ululyk file1-de kesgitlenip, file2-de we file3-de ulanylýan bolsa onda file2-de we file3-de *extern* beýan etmeklik hökmandyr, sebäbi hemme üç faýldaky functionlar şol bir daşky ululyga ýüzlenýärler. Praktikada hemme daşky ululyklaryň we functionlaryň beýan edilişini baş faýl diýilip atlandyrylýan (header-file) aýratyn faýlda ýazmak we programmanyň her bir faýlynyň başynda *#include* sözüniň kömegi bilen ony ýerleşdirmek amatly bolýar. Umumy şertleşik boýunça header file adynyň yzynda *.h* goýulýar. *<stdio.h>* header faýlda hem standart bibliotekanyň functionlary beýan edilýär. Header faýllar barada 4-nji bapda, standart biblioteka barada bolsa 7-nji bapda we B goşmaçada durulup geçiler.

Görnüşi ýaly getline we copy functionlarynyň ýörüteleşdirilen görnüşleriniň argumentleri ýok, şonuň üçin hem olaryň prototiplerini hem getline() we copy() ýaly ýazmak amatly ýaly bolup görünýär. Taze standart köne C programmalar bilen ylalaşygy saklar ýaly, argumentleriň bolmazlygyny degişli argumentleriň gabat gelyänligini barlamaly dældigi barada signal hökmünde kabul edýär. Şonuň üçin hem haçanda barlagy saklamaly we argumentleriň ýoklugyny aýyk görkezmek islese void sözünden peýdalanmanmaly. Biz bu meselä 4-nji bapda gaýdyp geleris.

Biz bu bölümde daşky ululyklar barada *kesgitlemek* hem-de *beýan etmek* düşüňjelerini örän seresaplylyk bilen ulanandygymyzy belläp geçmek gerek. “Kesgitlemek”-de ululyk döredilýär we oňa maşynyň huşunda ýer bölünip berilýär.

“Beýan” etmekde ululygyň tebigaty fiksirlenýär, ýöne onuň üçin hiç-hili ýer bölünip berilmeýär.

Hemme ululyklary daşky ululyk görnüşinde beýan etmeli diýen pikiriň hem bardygyny belläp geçmek gerek. Ilkinji gezek göräýmäge baglanyşyk ýönekeýleşýän ýaly bolup görünýär – argumentleriň toplумы gysga, ululyklara gerek bolan islendik ýerden ýüzlenip bolýar; ýöne ol ululyklar gerek däl ýerinde hem elýeterli bolup galýar. Daşarky ululykary juda köp ulanmak maslahat berilmeýär, sebäbi maglumatlar boýunça baglylyk düşnüksiz bolup galýar, üýtgeýän ululyklar garaşylman durka we gizligin usullar bilen üýtgäp bilmegi mümkin. Bulardan başgada beýle programmalary kämilleşdirmek kyn bolýar. Şol sebäplere görä, iň uzyn setiri tapýan programmanyň ikinji görnüşü birinji görnüşinden ýaramazdyr, olardan başgada iki peýdaly functionlaryň içinde hasaplamak üçin takyk ululyklaryň ulanylmagy bilen umumylyklarynyň bozulmagy,

Biz C diliniň ýadrosy diýilip aýdyp bolaýjak zatlara seredip geçdik. Beýan edilen “kerpiçjikler” uly ölçegli peýdaly programmalary düzmek üçin ýeterlikdir, şonuň üçin hem siz biraz wagtlap okamaklygy bes edip, dürli programmalary düzseňiz örän oňat bolar. Biz aşakdaky ýumüşlarda, öňki ýumüşlara garanyňda birnäçe esse kyn bolan meseleleri hödürleýäris.

Ýumuş 1. Girizilýän tekstde tabulýasiýa literalaryny, gerek bolan boşluklar bilen çalşyran detlab programmasyny düzüň. Her bir tabulýasiýa islendik ýerden başlap kesgitli n sany orny eýeleýär. Bu ýerde n -i üýtgeýän ululygyň bahasy görnüşinde bermek amatlymy ýa-da atlandyrylan hemişelik görnüşnde?

Ýumuş 2. Çapa çykarylýan tekstiň görnüşü üýtgemez ýaly, boşluklaryň zyzgiderligini iň az mukdardaky tabylýasiýa literleri we boşluklar bilen çalyşýan enlab programmasyny düzüň. Edil detlab programmasyndaky ýaly bir tabulýasiýa literasynyň näçe boşluk bilen çalşyrylmalydygyna üns beriň.??

Ýumuş 3. Giriş akymyndaky literalary n -nji orundan saga geçmez ýaly edip çapa çykarmak üçin programma düzmeli. Munuň özi, uzynlygy n -den geçýän setirlerleriň soňky bölegi indiki setire geçirilip çapa çykarylmaladygyny aňladýar. Setiriň indiki setire geçirmeli bölegini n -nji orundan çepde, iň soňky boşluk däl literadan soň “gözlemel”. Düzen programmaňyzyň örän uzyn setirler bilen, şeýle-de n -nji orna çenli hiç-hili boşluk, tabulýasiýa bolmadyk setirler bilen hem dogry işlemegi üçin alada ediň.

Ýumuş 4. Islendik C programmadan hemme düşüdirişleri (kommentariýalary) aýyryan programma ýazyň. Literleriň stringini we string hemişeliklerini bilen işläniňizde ünsli boluň. C dilinde düşündirişler iç-içine goýulmaýar.

Ýumuş 5. C programmadaky skobkalaryň hemme görnüşiniň balansyny (jübütligini) barlamak bilen ýönekeýje sintaksis barlag geçirýän programma ýazmaly. Dyrnaklar (goşa hem-de ýeke), eskeýp zyzgiderlikler (\...) we

kommentariýalar barada hem ýatdan çykarmaň. (Hemme ýagdaýlary gözöňünde tutup ýazjak bolsak, onda - bu çylşyrymly programmadar)

Tipler, operatorlar we aňlatmalar

Programmanyň iş salyşýan esasy obýektleri üýtgeýän ululyklar we hemişeliklerdir. Üýtgeýän ululyklar beýan edilişde sanalyp geçilýär, olaryň tipleri kesgitlenýär. Beýan edilişde üýtgeýän ululyklaryň başlangyç bahalary hem görkezilip biliner. Operasiýalar olaryň üstünden dürli täsirleri geçirýärler. Täze bahasyny almak üçin, aňlatma üýtgeýän ululyklaryň hem-de hemişelikleriň üstünde dürli operasiýalary geçirip biler. Obýektiň tipini onuň kabul edip biljek bahalarynyň we olaryň üstünden geçirilip bilinjek operasiýalaryň köplügi kesgitleýär. Bu babda şol ady agzalan “kerpiçjagazlar” barada gürrüň ederis.

ANSI standarty tarapyndan esasy tiplere we aňlatmalara ujypsyz üýtgetmeleriň we goşmaçalaryň köplügi tassyklandy. Indi islendik bitin tip alamatly (**signed**) we alamatsyz, (**unsigned**) bolup biler. Alamatsyz konstantalaryň we onalýlyk liter konstantalarynyň ýazgylary gözöňünde tutuldy. Ýüzýän nokatlar bilen geçirilýän operasiýalar indi birleýin takyklykda hem bolup biler. Ýokarlandyrylan takyklygy üpjün edýän **long double** tipi girizildi. Indi string ululyklar kompilyasiýa wagtynda konkatenirlenýär (“birleşdirilýär”). Tipiň belli bir aralygynda gurnalýan sanalýan tipler C diliniň bir bölegi bolup galdy. Bolup biläýjek tötänleýin üýtgetmelerden goramak maksady bilen obýektleri `conts` sözi bilen bellemeklige rugsat berilýär. Täze tipleriň girizilmegi bilen baglanyşyklykda, bir arifmetiki tipden beýlekisine awtomatiki öwürmekligi düzgüni giňeldildi.

Üýtgeýän ululyklaryň atlary

Biz 1-nji babda bu barada hiç zat aýtmadyk hem bolsak, üýtgeýän ululyklaryň we atlandyrylan hemişelikleriň atlarynda käbir çaklendirmeler bar. Atlar harplardan hem-de sanlardan ybarat bolup bilýärler, özünem birinji litera hökman harp bolmalydyr. Aşaky çyzyk belligi “_” harp hasaplanylýar; ony käbir uzyn atlar ýazylanda okamaklyk düşnükli bolar ýaly edip ulanmaklyk amatly bolýar. Üýtgeýän ululyklaryň atlaryny “_” literasy bilen başlamak maslahat berilmeýär, sebäbi biblioteka programmalarynyň köpüsy hut şol litera bilen başlanýar. Uly (baş) we kiçi (setir) harplar biri-birinden tapawutlanýarlar, ýagny `x` we `X` dürli atlardyr. Adat bolşy ýaly C programmada kiçi harplar bilen üýtgeýän ululyklar, uly harplar bilen bolsa atlandyrylan hemişelikler bellenilýär.

Içki atlar üçin ilkinji 31 literanyň ahmiýeti bardyr. Functionlaryň we daşky ululyklaryň atlary üçin bolsa 31 literadan az hem bolup biler, sebäbi ol atlar assembler we ýükleýjiler tarapyndan ulanylýar, dil tarapyndan olar barlanylmaýar. Daşky atlaryň unikallygyna(?) haýsy registerdeligine seretmezden diňe ilkinji 6 literiň çäginde güwä geçilýär. Diliň özünde ulanylýan sözler bolan `if`, `else`, `int`, `float`, we ş. m. eldegrilmesiz bolup, olar üýtgeýän ululyklaryň atlary hökmünde ulanmak bolmaýar. Olaryň hemmesi kiçi registerde ýazylýar(ýagny kiçi harpda).

Elbetde üýtgeýän ululyklara olaryň niýetlenişi bilen baglylykda biri-birinden tapawutly manyly atlaryň berilmegini maslahat berýäris. Biz lokal ululyklar üçin,

esasanam gaýtalanýan gurluşlaryň sanajýlary üçin gysgajyk atlaryň, has uzyn atlary bolsa daşky ululyklar üçin ulanmaklygy amatly hasaplaýarys.

Maglumatlaryň tipleri we ölçegleri

C dilinde bary ýogy birnäçe baza tipa bar:

char	bir baýt bolup, ol ýol berilýän literalardan diňe birini saklap bilýär.
int	bitin san bolup ol ulanylýan maşyndaky bitin sanlaryň tebigy ölçegini görkezýär (?).
float	ýüzýän nokatly, birleýin takykly san.
double	ýüzýän nokatly, ikileýin takyklykdaky san.

Bulardan başgada birnäçe kwalifikatorlar bolup, olary görkezilen baza tipleri bilen birlikde ulanmak bolýar. Mysal üçin, short (gysga) we long (uzyn) kwalifikatorlary bitin tipler üçin ulanylýar:

short int sh;

long int counter;

Beýle görnüşli beýan etmelerde adaty edilişi ýaly int sözünü görkezmän hem bolar.

Bitin short we bitin long dürli uzynlykly bolmaly, a int bolsa ulanylýan maşyndaky bitin sanyň tebigy ölçegine gabat gelýär. Köplenç ýagdaýda short kwalifikatory bilen ýazylan bitin sanlar üçin 16 bit , long kwalifikatory bitin sanlar üçin bolsa 32 bit ýer berilýär, int tipli ululyklar üçin bolsa 16 ýa-da 32 bit berilýär. Kompilýatory döredijiler ulanylýan kompýuterleriň karakteristikasyňa baglylykda, aşakdaky çäklendirmeleri gözönünde tutmak bilen bitin tipleriň ölçeglerini özlari saýlap bilerler: short we int tipleriň ölçegleri azyndan 16 bit, long tipiniňki bolsa 32 bit, short-yň ölçegi int-iň ölçeginden uly bolmaly däl, int-iň ölçegi bolsa long-yň ölçeginden uly bolmaly däl.

signed (alamatly) ýa-da unsigned (alamatsyz) kwalifikatorlaryny char ýa-da islendik bitin tipler üçin ulanmak bolar. unsigned tipine degişli ululygyň bahasy elmydama položitel ýa-da nula deňdir we moduly boýunça 2^n arifmetiki kanyna boýun egýär, bu ýerde n tipiň ikilik ýazgysyndaky bitleriň mukdary. Mysal üçin char baha üçin 8 bit berilýän bolsa, onda unsigned char -128... +127 bahalary kabul eder (maşynda ikilik doldurgyç kodlary). Ýöne char tipiniň alamatly ýa-da alamatsyzlygy maşyna bagly, ýöne islendik ýagdaýda hem çap edilýän literleriň kodlary položiteldir.

long double ýokarlandyrylan takyklykdaky ýüzýän nokatly arifmetika üçin niýetlenendir. Bitin sanlarda bolşy ýaly, ýüzýän nokatly obýektleriň ölçegleri hem kompilýatora baglydyr; float, double we long double şol bir ölçegli ýa-da iki ýa-da üç dürli ölçegli hem bolup bilerler.

Islendik ölçegli atlandyrylan hemişelikler maşynyň we kompilýatoryň beýleki karakteristikalary bilen birlikde standart baş <limits.h> we <float.h> faýllarda saklanýar. (B goşmaça serediň)

Ýumuş 1. Standart baş faýllardan degişli bahalary çap etmek we hasaplamak arkaly signed şeýle hem unsigned görnüşde ýazylan char, short, int we long tipleriň

kabul edip bilýän bahalarynyň çäklerini berýän programma düzmeli. Dürli tipli ýüzýän nokatly sanlaryň çäklerini kesgitleň. Bu çägi kesgitlemek bitin sanlaryňka garanyňda çylşyrymlydyr.

Hemişelikler

Bitin hemişelik, mysal üçin 1234 int tipine degişlidir. long tipli konstanta l ýa-da L bilen tamamlanýar, nysal üçin 123456789L; int tipi bilen görkezip bolmaýan juda uly bitin sanlar, long tipi arkaly beýan edilýärler. Alamatsyz hemişelikler u ýa-da U literalar bilen tamamlanýarlar, eger ul ýa-da UL bilen tamamlanýan bolsa onda hemişeligiň tipi - unsigned long.

Ýüzýän nokatly hemişelikleriň onluk nokady (123.4) ýa-da eksponensial bölegi (1e-2) bolýar ýa-da olaryň ikisi hem bolýar. Eger olaryň soňlamasy (окончания) bolmasa onda olar double tipine degişli hasaplanylýar. f we F soňlamalar float tipine degişlilik görkezýär, a l we L soňlamalar bolsa long double tipine degişlilik görkezýär.

Bitin onluk sanlardan başgada bitin sekizlik we bitin onaltlyk sanlar hem bolup bilerler. Eger hemişelik nuldandan başlanýan bolsa, onda ol sekizlik görnüşde ýazylypdyr, egerde hemişelik 0x-den ýa-da 0X başlanýan, bolsa onda ol onaltlykda ýazylypdyr diýilip hasaplanylýar. Mysal üçin onluk bitin san bolan 31-i 037-i ýa-da 0X1F görnüşde ýazmak bolar. Sekizlik we onaltlyk hemişelikleriň ýazgysy L soňlama (bu ýerde L long tipi görkezýär) ýa-da U (hemişelik alamatsyz, unsigned) bilen tamamlanyp biler. Mysal üçin 0XFUL 15-e deň bolup unsigned long tipe degişli bolar.

Liter hemişelikler ‘x’ görnüşdäki ýaly birleýin dynnagynyň içinde ýazylyan literadan ybarat bitin ululykdyr. Onuň san bahasy, berlen maşyndaky literleriň ýygynyndasyndaky literiň kodyna deňdir. Mysal üçin ASCII kodirowkada ‘0’ hemişelik 48-e deň, görnüşi ýaly onuň 0 san bilen hiç-hili dahyly ýokdyr. Eger biz haýsydyr bir baha (mysal üçin 48) ýazman ‘0’ ýazsak, onuň san bahasy kodirleniş usulyna laýyklykda kesgitlener, ýöne ol ýazgy liter ululyk görnüşinde kodirmek usulyna bagly däl we adaty sandan has düşnükli. Liter hemişelikler, köplenç ýagdaýda beýleki literler bilen deňeşdirmek maksady bilen ulanylýan hem bolsa, olar islendik beýleki bitin tipler ýaly, sanlaryň üstünden geçirilýän operasiýalarda hem ulanylyp biliner.

Birnäçe literler, liter we setir hemişeliklerinde Esc yzygiderligiň kömegi bilen ýazylyýarlar, mysal üçin \n (täze setir); beýle yzygiderlikler iki sany literiň kömegi bilen ýazylyýan hem bolsalar, olar bir literi aňladýar. Mundan başgada islendik sekizlik kody aşakdaky ýaly ýazmak bolar

‘\000’

bu ýerde 000 – bir, iki ýa-da üç sekizlik san(0...7) ýa-da

‘\xhh’

bu ýerde hh – bir , iki ýa-da köp onaltlyk sanlar(0...9, a...f, A...F). Şeýlelikde biz aşakdaky görnüşde ýazyp bileris

#define VTAB ‘\013’

/* ASCII-däki wertikal tabulýasiýa */

#define BELL ‘\007’

/* ASCII-de jaň */

ýa-da onaltylyk görnüşi:

```
#define VTAB '\xb'          /* ASCII-däki wertikal tabulýasiýa */
#define BELL '\x7'          /* ASCII-de jaň */
```

Esc yzygiderlikleriň doly toplumy aşakdakylardan ybaratdyr:

\a	jaň-sinaly	\\	ters ýapgyt çyzyk
\b	bir ädim yza	\?	sorag belgisi
\f	sahypany geçirmek	\'	birleýin dyrnak
\n	täze setir	\"	goşa dyrnak
\r	karetkany yzyna gaýtarmak	\000	sekizlik kod
\t	kese tabulýasiýa	\xhh	onaltylyk kod
\v	dik tabulýasiýa		

'\0' litera belgisi – ol nul bahaly litera, ýa-da aýdylyşy ýaly null litera. Aňlatmanyň litera tebigatyny görkezmek üçin 0 -yň ýerine '\0' ýazýarlar, aslynda ol ý-da beýleki ýagdaýda hem ýazgy nuly aňladýar.

Aňlatma hemişeligi – diňe konstantalaryň üstünden operasiýa geçirýän aňlatmadyr. Beýle aňlatmalaryň bahasy hasaplama wagtynda dälde kompilýasiýa döwründe hasaplanylýar, şonuň üçin hem olary hemişelikleriň ulanylýan islendik ýerinde ulanyp bolar, mysal üçin

```
#define MAXLINE 1000
char line[MAXLINE+1];
```

ýa-da

```
#define LEAP 1 /* in leap years – uzak ýyllar */
int days[31+28+LEAP+31+30+31+30+31+31+30+31+30+31];
```

Setir hemişeligi ýa-da setir literalý – goşa dyrnagyň içinde ýazylan nul ýada ondan köp bolan literalardyr, mysal üçin

“Bu setir hemuşeligi”

ýa-da

```
“” /* boş setir */
```

Goşa dyrnaklar setiriň içine girmeyär, olar diňe setiriň çäginde görkezýär. Liter hemişeliklerindäki ýaly, setir hemişelikleriniň düzümine Esc yzygiderlikleri hem goşmak bolar; mysal üçin \" goşa dyrnagy aňladýar. Setir hemişeliklerini kompilýasiýa döwründe konkatenirmek (birleşdirmek) bolýar; mysal üçin aşakdaky iki setiriň ýazgysy

“Salam” “Dünýe!”

aşakdaky bir setiriň ýazgysyna deňdir:

“Salam Dünýe!”

Setirleriň bu häsiýeti uzyn setirleri böleklere bölmäge we bu bölekleri aýratyn setirde saklamaga mümkinçilik berýär.

Setir hemişeligi – munuň özi literalaryň massiwidir. Setir maşynyň ýadynda ýerleşende onuň ahyrynda hökman '\0' litera durýar, şol sebäpli setir üçin niýetlenen kompýuteriň ýady onuň öz uzynlygyndan bir baýt köp ýer tutýar. Bu

döredilýän setiriň uzynlygyna çäk goýulmaýanlygyny aňladýar, ýöne onuň uzynlygyny kesgitemek üçin tutuş setire seredip geçmek zerurdyr. `strlen(s)` `s` setiriň iň soňky `'\0'` literasyny göz önünde tutmazdan, onuň uzynlygyny hasaplaýar. Aşakda ol functionyň biziň tarapymyzdan ýazylan programmasyny getirýäris:

```
/* strlen: s setiriň uzynlygyny kesgitleýär */
int strlen(char s[])
{
    int i;
    i = 0;
    while (s[i] != '\0')
        ++i;
    return i;
}
```

`strlen` we setirler üçin niýetlenen beýleki functionlar `<string.h>` standart baş faýlda ýazylandyr.

Uns beriň we ýatda saklaň, bir literany saklaýan liter ululyk bilen bir literany saklaýan setir ululyk şol bir zat däldir: `'x'` bilen `"x"` şol bir zat däldir. `'x'` ýazgy `x` literanyň kodyna deň bolan bitin sandyr, `"x"` ýazgy bolsa literalaryň massiwi bolup ol häzirki wagtda bir literany (`x` harpy) we `'\0'`-y özünde saklaýar.

C dilinde hemişelikleriň ýene bir görnüşi, sanalýan hemişelikler ulanylýar. Sanalýan hemişelikler – bitin hemişelikleriň aşakdaky ýaly

```
enum boolean {NO, YES}
```

sanalýan görnüşidir. `enum` (sözün manysy - sanalmak)-daky birinji at 0 baha eýe, indili 1, we ş.m. (eger hemişelikleriň bahalary aýyk görkezilmedik bolsa). Eger hemme hemişelikleriň bahalary görkezilmedik bolsa, onda aşakdaky iki mysaldaky ýaly, iň soňky görkezilen bahadan başlap artmaklygy (prograssiýa) dowam etdirýärler:

```
enum escapes {    BELL = '\a', BASCSPACE = '\b', TAB = '\t',
                  NEWLINE = '\n', VTAB = '\v', RETURN = '\r'};
```

```
enum months {    JAN = 1, FEB, MAR, APR, MAY, JUN,
                  JUL, AUG, SEP, OCT, NOV, DEC };
/* FEB 2-ä, MAR 3-e deň, we ş.m. */
```

Atlar dürli sanalmalarda biri-birinden tapawuplanmalydyrlar. Şol bir sanalmanyň içinde bahalar gabat gelip bilerler.

`enum` serişdesi hemişeliklere at dakmaklygyň amatly usulydyr, özünem `#define`-den tapawutlylykda hemişelikleriň bahalary awtomatik döredilip hem biliner. Sanalýan tipleri üýtgeýän ululyklary kesgitlemek üçin ulanmaklyga ygtyýar berilýär, ýöne ol ululyklara dakylýan baha olaryň tipine gabat gelýänligini barlamak kompilýatoryň wezipesine girmeýär. Ýöne diňe, şeýle mümkinçiligiň barlygynyň özi hem, `enum`-yň `#define`-den gowdygyny goörkezýär. Bulardan başgada `debug` (отладчик) sanalýan ululyklaryň bahalaryny simwol görnüşde çap etmäge mümkinçilik alýar.

Deklarirlemek

Hemme ululyklar özleriniň ulanylmazyndan öň deklarirlenmelidirler, özünem käbir deklarasiýalar aýyk däl görnüşde berilip bilener. Deklarirlemek munuň özi tipi, hemde şol tipe degişli bolan üýtgeýän ululyklaryň birini ýa-da toplumyny görkezmekdir, mysal üçin:

```
int lower, upper, step;
```

```
char c, line[1000];
```

Üýtgeýän ululyklar erkin görnüşde dürli hili deklarirlenip biliner, mysal üçin ýokardaky deklarirlenmekligi aşakdaky görnüşde hem ýazmak bolar:

```
int lower;
```

```
int upper;
```

```
int step;
```

```
char c;
```

```
char line[1000];
```

Deklarirlemekligiň soňky görnüşi köp ýer alýan hem bolsa, ol gowydyr, sebäbi her bir deklarirlenen ululyga kommentariý (düşündiriş) ýazmaga mümkinçilik berýär we gelejekgi kämilleşdirmek üçin has amatlydyr.

Üýtgeýän ululyk deklarirlenýän döwründe onuň başlangyç bahasy görkezilip hem biliner, mysal üçin:

```
char esc = '\\'
```

```
int i = 0;
```

```
int limit = MAXLINE+1;
```

```
float eps = 1.0e-5;
```

Awtomatik däl üýtgeýän ululyklaryň inisializasiýasy (başlangyç bahany dakmak) diňe bir gezek, programma ýerine ýetip başlamazyndan öň ýerine ýetirilýär, özünem onuň inisializatory hemişelik aňlatma bolmalydyr. Aýyk inisializirlenýän awtomatik üýtgeýän ululyklar her gezek functiona ýa-da bloga girilende başlangyç bahany alýar, onuň inisializatory islendik aňlatma bolup biler. Daşky we statik üýtgeýän ululyklaryň başlangyç bahasy görkezilmedik halatynda nula deň bolýar. Aýyk görnüşde inisializirlenmedik awtomatik üýtgeýän ululyklar kesgitsiz bahalary saklaýarlar (“musor”, “zibil”).

Deklarasiýada Islendik üýtgeýän ululyk üçin, onuň bahasynyň soňra üýtgemeyänligini görkezmek maksady bilen const kwalifikator ulanylyp biliner.

```
const double e = 2.71828182845905;
```

```
const char msg[] = “duýduryş”;
```

const kwalifikatorly massiwler üçin ulanylanda, ol massiwiň hiç bir elementiniň üýtgedilip bilinmejekdigini aňladýar. const görkezmesi argument massiw üçin hem ulanylyp biliner, beýle ýagdaýda function ol massiwi üýtgedip bilmeýär:

```
int strlen(const char[]);
```

const kwalifikator bilen bellenen üýtgeýän ululygyň bahasynyň üýtgediljek bolundaky reaksiýa kompilyatorya baglydyr.

Arifmetiki operatorlar

$+$, $-$, $*$, $/$ şeýle hem modul almak $\%$ operatory binar arifmetiki operasiýalardyr. Bitin sanlar özara bölünen ýagdaýynda olaryň drob bölekleri taşlanylýar.

$x \% y$

aňlatma x ululyk y ululyga bölünendäki galyndyny kesgitleýär, eger-de ol tutuşlygyna bölünýän bolsa aňlatmanyň bahasy nula deň bolar. Mysal üçin eger ýyl 4-e galyndysyz bölünýän bolsa (ýöne 100- bölünmeýän bolsa) ol ýyl uzak ýyl hasaplanylýar. Mundan başgada 400-e bölünýän ýyllar hem uzak ýyllar hasaplanylýar. Şeýle ýazyp bileris

```
if ((year % 4 == 0 && year % 100 != 0 || year % 400 == 0)
```

```
    print("%d uzak ýyl\n",year);
```

```
else
```

```
    printf("%d adaty ýyl\n",year);
```

$\%$ operatory float we double tipli operandlar üçin ulanylmaýar. / ýerine ýetirilende drob böleginiň haýsy tarapa kesiljekdigini (artýan ýa-da kemelýän tarapyna) we $\%$ operatory otrisatel operandlar bilen ýerine ýetende netijäniň alamatynyň nähili boljakdygy bu nasýna bagly.

$+$ we $-$ binar operatorlaryň birmeňzeş prioritetleri (ýerine ýetiş tertibi) bar. Olaryň prioritetleri $*$, $/$ we $\%$ operatorlaryň prioritetlerinden aşakdayr, olaryň prioritetleri bolsa öz gezeginde $+$ we $-$ unar operatorlaryňkydan aşakdadyr. Şol bir prioritetli arifmetiki operasiýalar çepden başlap saga tarap ýerine ýetýärler.

Bu babyň soňunda ýerleşdirilen 2.1 tablisada hemme operatorlaryň prioritetleri we olaryň ýerine ýetiş tertipleri görkezilendir.

Gatnaşyk operatorlary we logiki operatorlar

Aşakdaky operatorlara gatnaşyk operatorlardyr

$<$ $>=$ $<$ $<=$

Olaryň hemmesiniň birmeňzeş prioritetleri bardyr.

$==$ $!=$

operatorlarynyň bolsa ýokarkylar bilen deňşdireniňde bir basgançak pes prioriteti bardyr. Gatnaşyk operatorlary arifmetiki operatorlara garanyňda pes prioritetlidir, şonuň üçin $i < \text{lim}-1$ aňlatma, biziň garaşyşymyz ýaly $i < (\text{lim}-1)$ aňlatma ýaly ýerine ýeter.

$\&\&$ we $\|$ logiki operatorlar has gyzyklydyr. Aralarynda $\&\&$, $\|$ operatorlar duran aňlatma çepden saga hasaplanylýar we haçanda netijäniň çynlygy ýa-da ýalandygy belli bolanda hasaplalamaklyk togtadylýar. Köp C programmalar bu häsiýeti ulanýarlar, mysal üçin 1-nji bapdaky getline functionyndaky gaýtalanmak:

```
for (i=0; i<lim-1 && (c=getlchar()) != '\n' && c != EOF; ++i)
```

```
    s[i] = c;
```

Indiki literany okamazdan öň, onuň üçin s massiwde ýer bardygyny ýa-da ýoklygyny barlamaly, başgaça aýdanymyzda $i < \text{lim}-1$ şerti ýerine barlamaly. Eger bu şert ýerine ýetmeýän bolsa, onda biz hasaplamany dowam etmeli dälde, ýagny indiki literany okamaly dälde. Şeýle hem getchar functiona ýüzlenmezden

öň c ululygy EOF bilen deňşdirmek nädogry bolardy; ýagny getchar functionyny çagyrmak we eýe bolmak ol barlagdan öň ýerine ýetirilmelidir.

&& operatorynyň möhümligi (prioriteti) || operatorynyňkydan ýokarydyr, ýöne olaryň möhümligi gatnaşyk we deňlik operatorlarynyňkydan pesdir; şonuň üçin hem

```
i<lim-1 && (c = getchar()) != '\n' && c != EOF
```

aňlatmada goşmaça skobkalar hokman däl. Ýöne != operatorynyň möhümligi eýe bolmak operatorynyňkydan ýokarydyr, şonuň görä

```
(c = getchar()) != '\n'
```

aňlatmada ilki bilen eýe bolmaklygyň, soňra bolsa '\n' bilen deňşdirmekligiň ýerine ýetmegi üçin skobka zerurdyr. Kesgitleme boýunça gatnaşyk ýa-da logiki aňlatmalaryň san bahasy eger olar çyn bolanda 1-e deň we yalan bolanda 0-a deň.

! unar operatory nul däl operandy 1-a we nul operandy bolsa 1-e öwürýär. Adatça ! operator

```
if (!valid)
```

ýaly konstruksiýalarda ulanylyp, ol

```
if (valid == 0)
```

ýazga ekwiwalentdir.

Haýsy görnüşdäki ýazgynyň gowydygyny aýtmak kyn. !valid görnüşdäki konstruksiýa ýeňil okalýar (“eger valid däl bolsa”), ýöne has çylşyrymly aňlatmalarda oňa düşünmekligiň kynyrak bolmagy hem mümkin.

Tipleri özgertmek

Eger operatoryň operandlary dürli tiplere degişli bolsa, onda olar käbir umumy tipe getirilýärler. Bu birnäçe düzgünleriň esasynda ýerine ýetirilýär. Adatça diňe hiç-hili maglumat ýitmezden kiçi diapazondaky bahaly operandlar, uly diapazondaky bahaly operandlara awtomatik görnüşde öwrülýärler, mysal üçin f + i görnüşdäki aňlatmalarda bitin sanyň ýüzýän nokatly sana öwrülmeği. Indeks görnüşde ýüzýän nokatly sanyň ulanylmagy ýaly manysyz aňlatmalara ýol berilmeýär. Maglumatlaryň ýitip biljek aňlatmalary (aýdaly uzyn bitin üýtgeýän ululyklary gysga tiplere dakylanda ýa-da ýüzýän nokatly ululyklaryň bahasyny bitin tiplere dakylanda) ýazylsa, bu barada duýduryjy habarlar berlip biliner, ýöne olar ýol bererlikdir.

char tipiniň bahasy – bu umuman kiçi bitin ululykdyr we olary arifmetiki aňlatmalarda erkin ulanmak bolar, bu bolsa literleriň üstünden geçirilýän mümkin bolan manipulýasiýalary juda ýeňilleşdirýär. Mysal hökmünde sanlaryň yzygiderligini olaryň san ekwiwalentine öwürýän atoi functionyň ýönekeýje görnüşde ýazylyşyna seredip geçeliň.

```
/* atoi: s-i bitin sana öwürmek */
```

```
int atoi(char s[])
```

```
{
```

```
    int i, n;
```

```
    n = 0;
```

```
    for (i = 0; s[i] >= '0' && s[i] <= '9'; ++i)
```

```
        n = 10 * n + (s[i] - '0');
```

```

        return n;
    }
Biziň 1-nji bapda aýdyşymyz ýaly
    s[i] – ‘0’

```

aňlatma s[i]-da saklanýän literanyň san bahasyny berýär, ýagny ‘0’, ‘1’ we ş.m. üznüksiz osýän yzygiderligi emele getirýär.

char tipi int tipe öwürmek lower functiony bilen baglanyşykly, ol ASCII kodlar ýygyndysyndan bir literany alýar we eger ol baş harp bolsa ony setir harpa öwürýär. Eger litera baş harp bolmasa onda lower ony üýtgetmeýär.

```

/* lower: c-ni setir harpa öwürmek; diňe ASCII üçin */

```

```

int lower(int c)
{
    if (c >= ‘A’ && c <= ‘Z’)
        return c + ‘a’ – ‘A’;
    else
        return c;
}

```

ASCII kodlar ýygyndysynda bu programma dogry işlär, sebäbi şol bir baş we setir harplaryň arasyndaky uzaklyk meňzeş (eger olara san hökmünde seretsek) we mundan başgada latyn harplary dykyz ýerleşen, ýagny bu kodlar ýygyndysynda A hem-de Z harplaryň arasynda diňe harplar ýerleşýär. EBCDIC kodlar ýygyndysynda ýokarky şert ýerine ýetmeýär, şonuň üçin hem biziň programmamyz bu ýagdaýda harplardan beýleki literalary hem üýtgeder.

B goşmaçada beýan edilen standart baş faýl <ctype.h> haýsy kodlar ýygyndysyndadygyna bagly bolmazdan literalary barlaýan we özgertýän functionlaryň maşgalasyny özünde saklaýar. Mysal üçin tolower© function, eger c uly registrde ýazylan bolsa ony kiçi registre öwürýär, şonuň üçin hem tolower functiony ýokarda seredilen lower functionyny islendik ýagdaýda çalyşýar.

```

c >= ‘0’ && c <= ‘9’

```

barlagy

```

indigit©

```

bilen çalşyrmak mümkin.

Indiden beýläk biz <ctype.h> -yň functionlaryny ulanarys.

Literi bitin sana öwürmekligiň bir kynçylygy bar: C dili char tipe degişli ululygyň alamatlydygyny ýa-da alamatly dældigini kesgitlemeýär. char-y int-e öwrenimizde haçandyr bir wagt otrisatel bitin san alnyp bilermi? Dürli arhitekturaly maşynlar üçin dürli hili jogaplaryň bolmagy mümkin. Käbir maşynlarda uly biti birlik bolan char tipli ululyklar otrisatel bitin sanlara öwrülýärler (“alamatlarynyň köpelmegi” serişdesi boýunça). Beýleki özgertmelerde bolsa char-y int-e öwürmek üçin onuň ikilik bitleriniň çepinden bir nuly goşulýar we elmydama položitel san alynýär.

Çap edilýän literleriň ýygyndysyndaky islendik litera otrisatel bolup bilmejekligine garantiýa berilýär, şonuň üçin aňlatmalarda şeýle literalar elmydama položitel operandlaryň bolup gelýärler. Ýöne sekizbitlik (bitli islendik bir kod) char tipli üýtgeýän ululyk bir maşynda otrisatel san, beýleki bir maşynda

bolsa položitel san bolup biler. Ylalaşyklygyň saklanmagy üçin liter däl ululyklary hem saklap bilýän char ululyklary beýan edilende signed ýa-da unsigned kwalifikatirlary ulanmak maslahat berilýär.

&& we || operatorlar bilen bilelikde ulanylýan $i > j$ tipli gatnaşyk we logiki aňlatma çyn bolsa 1 we ýalan bolsa 0 aňlatma-şerti kesgitleýärler. Şeýlelikde

$d = c >= '0' \&\& c <= '9'$

eýe bolmakda c ululyk san bolsa onda d ululygyň bahasy 1 bolar we c ululyk san bolmadyk ýagdaýynda 0 bolar. Ýöne netije çyn bolan ýagdaýynda isdigit ýaly functionlar nuldan tapawutlanýan islendik bahany berip biler. if, while, for we ş.m.-de barlag geçirilende, şert nula deň däl bolsa ol çyn hasaplanylýär.

Açyk däl arifmetiki özgertme düzgün bolşy ýaly tebigy şekilde amala aşyrylýar. Umumy ýagdaýda iki operandly (binar operatorlar) + ýa-da * tipli operatorlar dürli alamatly operandly bolsalar, onda operasiýa başlamazdan ön kiçi tipler “ösyärler” we “uly” tiplere öwrülýärler. Öwürmek düzgünü takyk kesgitlenendir. Eger aňlatmada alamatsyz operandlar ýok bolsa onda aşakdaky formal däl düzgünler bilen çäklenmek bolar:

- Eger haýsy hem bolsa bir operand long double tipe degişli bolsa onda beýleki hem long double tipe getirilýär.
- Ýokardaky şert ýerine ýetmedik ýagdaýynda, operandlaryň biri double tipe degişli bolsa onda beýleki operand hem double tipe getirilýär.
- Ýokardakylar ýerine ýetmedik ýagdaýynda, operandlaryň biri float tipe degişli bolsa onda beýleki operand hem float tipe getirilýär.
- Ýokardaky şertler ýerine ýetmedik ýagdaýynda char we short tipli operandlar int tipe getirilýär.
- we iň soňunda operandlaryň biri long tipli bolsa onda beýleki hem long tipe getirilýär.

float tipli operandlaryň awtomatik suratda double tipli operandlara getirilmeýänligine üns bereliň; şonuň bilen hem C diliniň bu wersiýasy ilki başdakysyndan tapawutlanýär. Umuman aýdanymyzda <math.h> bibliotekasya ýygňalan matematiki functionlar hasaplamalarda ikeldilen takyklykda işlemek üçin niýetlenendir. float esasan uly massiwlerde maşynyň ýadyny tygşytlamak maksady bilen we kähallatlarda ikileýin takyklykdaky arifmetika gymmat düşýän maşynlarda hasaplamany çaltlaşdyrmak üçin ulanylýär.

unsigned operandlaryň ýüze çykmagy bilen öwürmek düzgünü çylşyrymlaşýar. Esasy kynçylyk alamatly we alamatsyz bahalaryň deňeşdirilmegi bitin tipiň ölçegine bagly, bitin tipiň ölçeginiň bolsa dürli maşynlarda dürli hili bolmagy mümkin. Goý int tipiň bahasy 16 bit, a long tipiniň bahasy bolsa 32 bit ýer tutýar diýeliň. Onda $-1L < 1U$, sebäbi 1U int tipine degişli we signed long tipine özgerdilýär we uly položitel san hökmünde kabul edilýär.

Özgerdilmek eýe bolmakda hem bolup geçýär: Eýe bolmaklygyň sag böleginiň tipi çep böleginiň tipine getirilýär, ol bolsa netijäniň tipi bolýar.

Litera alamatlarynyň köpeldilmegi arkaly ýa-da ýokarda beýan edilen beýleki usullaryň biri arkaly bitin sana öwrülýär.

Uly razryadlarynyň taşlanmagy bilen uzyn bitin sanlar kiçi bitin sanlara ýa-da char tipe öwrülýär. Şeýlelikde aşakdaky programma böleginde

```
int i;  
char c;  
i = c;  
c = i;
```

c ululygyň bahasy üýtgemez. char tip int tipe öwrülende, alamatlaryň köpeldilmegine bagly bolmazdan elmydama dogrydyr. Ýöne eýe bolmaklygyň tertibini üýtgetsek onda maglumatyň ýitmegi mümkin.

Eger x float tipe degişli bolsa, a i int tipe degişli bolsa, onda $x = i$, $i = x$ instruksiýalar tipleriň özgerdilmegine getirer, özüňem float tipi int tipe drop böleginiň taşlanmagy bilen amala aşyrylýär. Eger double tipi float tipine öwrülýän bolsa onda ýaha tegeleklenýär, ýa-da kesilýär; bu işiň ýerine ýetiriliş usulyna (realizasiýasyna) bagly.

Ýagny function çagyrylandaky argument aňlatmadyr, ol functiona geçirilende hem tipleriň özgerdilmesiniň bolmagy mümkin. functionyň prototipiniň bolmadyk ýagdaýynda char we short tipli argumentler int tipine öwrülýärler, float tipi bolsa double tipine öwrülýärler. Şonuň üçin biz haçanda functionlary çagyranymyzda char ýa-da float tipli argumentleri ulanan hem bolsak, ol argumentleri int ýa-da float görnüşde beýan etdik.

Islendik aňlatma üçin unar operatory ulanmak bilen onuň tipiniň özgerdilmegini aýyk görkezmek bolar. Aşakdaky ýaly

(tipiň-ady) aňlatma

ýazgy ýokarda sanalyp geçilen düzgünler boýunça aňlatmany skobkanyň içindäki görkezilen tipe getirýär. Getirmek operasiýasynyň manysyny şeýleräk gözöňüne getirmek bolar: aňlatma görkezilen tipli käbir üýtgeýän ululyga eýe bolýar we bu üýtgeýän ululyk bütün konstruksiýanyň ýerine ulanylýar. Mysal üçin bibliotekadaky sqrt programma double tipli argumentler üçin niýetlenendir we onuň argumentinde başga tipli argument görkezilse ol function manysyz zatlary yzyna gaýtaryp berer. (sqrt <math.h>-da beýan edilendir.) Şonuň üçin hem eger n bitin ululyk bolsa, onda biz aşakdaky ýaly ýazyp bileris

```
sqrt((double) n)
```

we n-iň bahasy functiona berilmezinden öň onuň tipi double tipine öwürüler. Bu ýerde n-iň bahasy double tipinde döredilýär, ýöne n-iň asul bahasy üýtgemeýär. Getirmek operatorynyň möhümligi (prioriteti) bu baby soňunda ýerleşdirilen tablisadan görnüşi ýaly islendik unar operatorlaryňkydan ýokarydyr.

Haçanda argumentler functionyň prototipinde beýan edilen bolsa, garaşylyşy ýaly, functionlar çagyrylanda gerekli tip özgertermeleri awtomatik bolup geçýär. sqrt functionyň aşakdaky ýaly

```
double sqrt(double);  
prototipi bolan ýagdaýynda  
root2 = sqrt(2);
```

eýe bolmaklykda sqrt-ä ýüzlenilmezinden öň bitin 2 san aýyk geçirmek operasiýasyny görkezmezden awtomatik suratda double tipine (2.0) öwürüler.

Geçirmek operasiýasyny, standart biblioteka girýän tötänleýin san generatorynyň erkin wersiýasynda we functionda göreliň.

```
unsigned long int next = 1;
/* rand: tötänleýin 0..32767 aralykdaky bitin sanlary kesgitleýär */
int rand(void)
{
    next = next * 1103515245 + 12345;
    reurn (unsigned int) (next/65536) % 32768;
}

/* srand: rand üçin “zatrawka”-ny goýýar */
void srand(unsigned int seed)
{
    next = seed;
}
```

Ýumuş 1. 0x ýa-da 0X bilen başlanýan onaltylyk san yzygiderligini degişli bitin sana öwürýän htoi(s) functionyny ýazyň. 0..9, a...f, a...F literlere onaltylyk sanlar diýilýär.

Inkrement we dekrement operatorlar

C dilinde üýtgeýän ululyklaryň bahasyny artdyrmak we kemeltmek üçin niýetlenen iki sany adaty bolmadyk operator bar. Inkrement operator ++ öz operandyna 1-i goşýar, a dekrement operator -- bolsa 1-i aýyrýar. Biz eýýäm birnäçe gezek üýtgeýän ululyklaryň bahasyny artdyrmak üçin ++ operatorndan peýdalandyk, mysal üçin

```
if (c == '\n')
    ++nl;
```

++ we -- operatorlara adaty däldigi, olary prefiks operatorlary (üýtgeýän ululygyň öňünden ýeleşdirmek ++n) we postfiks operatorlary (üýtgeýän ululygyň soňunda ýerleşdirmek n++) görnüşlerde ulanmak bolýanlygy üçindir. Ýöne ++n n-iň bahasynyň ulanylmazyndan öň onuň bahasyny ulaldýär, a n++ bolsa onuň bahasy ulanylandan soň onuň bahasyny ulaldýar. Goý n 5-i saklaýan bolsun, onda

```
x = n++;
```

x ululyga 5-i başi bahany dakar,

```
x = ++n;
```

x ululygyň bahasy 6-y bolar. Ol ýa-da beýleki ýagdaýda hem x-iň bahasy 6-a deň bolar. Inkrement we dekrement operatorlary diňe üýtgeýän ululyklar üçin ulanmak bolar. Mysal üçin, (i+j)++ ýazgy nädogrydyr.

```
if (c == '\n')
    nl++;
```

ýaly, diňe üýtgeýän ululygyň bahasyny kiçeltmek ýa-da ulaltmak üçin prefiks ýa-da postfiks operatorynyň haýsysyny ulananymyzyň tapawudy ýokdyr. Ýöne käbir ýagdaýlarda bu operatorlaryň diňe birini ulanmaly bolýar. Mysal üçin s stringden c bilen gabat gelýän hemme literalary aýyrýan squeeze(s,c) functionyna seredip geçeliň:

```

/* squeeze: s-den hemme c-leri aýyrýar */
void squeeze(char s[], int c)
{
    int i,j;
    for (i= j = 0; s[i] != '\0'; i++)
        if (s[i] != c)
            s[j++] = s[i];
    s[i] = '\0';
}

```

Haçanda c-den tapawutlanýan litera gabat gelende ol j-nji oruna kopirlenýär we diňe şondan soň j-iň bahasy 1 san galdyrylýar, ýagny indiki literanyň kabul edilmegine taýýarlanylýar. Bu aşakdakylar bilen takyk gabat gelýär:

```

if (c == '\n') {
    s[i] = c;
    ++i;
}

```

has tygşytly hem ýazmak bolar:

```

if (c == '\n')
    s[i++] = c;

```

Üçünji mysal hökmünde t stringi s stringiň ahyrynda ýerleşdirýän strcat(s,t) standart functionyna seredeliň. s stringde t stringi hem özünde saklap biljek derejede orun bar diýeliň. Biz strcat functionyny hiç-hili netijäni gaýtaryp bermez ýaly edip ýazdyk. Bibliotekadaky strcat functiony bolsa netijeleýji stringe ýüzlenmäni (pointer, ссылка) gaýtaryp berýär.

```

/* strcat: t-ni s-iň soňunda ýerleşdirýär; s geregiçe uly */
void strcat(char s[], char t[])
{
    int i,j;
    i = j = 0;
    while (s[i] != '\0') /* s-iň soňuny tapýarys */
        i++;
    while ((s[i++] = t[j++]) != '\0') /* t-ni kopirleýäris */
        ;
}

```

t-den s-e indiki litera kopirlenende i we j üçin ++postfiks operator ulanylmagynyň sebäbi, gaýtalanmaklygyň her bir ädiminde i we j ululyklaryň geçirilýär literanyň ornuny dogry kesgitlemekleri üçindir.

Geronyň formulasy boýunça üçburçlygyň meýdanyny hasaplaýan programma:

```

#include <stdio.h>
#include <conio.h>
#include <math.h>
int main(void)
{
    clrscr();
}

```

```

float x1,y1,x2,y2,x3,y3,a,b,c,p,s;
printf("x1="); scanf("%f",&x1);
printf("y1="); scanf("%f",&y1);
printf("x2="); scanf("%f",&x2);
printf("y2="); scanf("%f",&y2);
printf("x3="); scanf("%f",&x3);
printf("y3="); scanf("%f",&y3);

a=sqrt(pow(x2-x1,2)+pow(y2-y1,2));
b=sqrt(pow(x3-x1,2)+pow(y3-y1,2));
c=sqrt(pow(x3-x2,2)+pow(y3-y2,2));

p=(a+b+c);
printf("Perimetr = %5.2f\n",p);
p=p/2;

s=sqrt(p*(p-a)*(p-b)*(p-c));

printf("Meydany = %5.2f",s);

getch();
return 0;
}

```

Ýumuş 1. s2 strinde gabat gelýän literalaryň hemmesini s1 stringden aýyryň, squeeze(s1,s2) functionyň bir wersiýäsyny ýazmalyň.

Ýumuş 2. Eger s1 setir ululygyny düzýän literalaryň içinden, s2 setir ululygyndaky literalaryň biri bilen gabat gelýäni bar bolsa, onda onuň duran ornuny çapa çykarýän ýa-da -1-i (minus 1-i) gaýtaryp berýän (s1-iň hiç bir literasy s2-iň literalary bilen gabat gelmeýän bolsa) any(s1,s2) functiony ýazyň. (Standart bibliotekadaky strpbrk functiony edil şol işi ýerine ýetirýär, ýone literanyň nomerini dälde, görkezijisini yzyna gaýtaryp berýär.)

Bitleýin operatorlar

C dilinde bitler bilen işlemäge mümkinçilik berýän alty sany operator bar. Olary diňe bitin tipli operandlar bilen ýagny alamatly ýa-da alamatsyz char, short, int we long tipler üçin ulanmak bolar.

&	bitleýin AND
	bitleýin OR
^	bitleýin öçüriji OR
<<	çepe süýşürmek
>>	saga süýşürmek
~	bitleýin inkär etmek (unar)

& operatory (bitleýin AND) razrýadlaryň toparynyň birnäçesini nula öwürmek üçin ulanylýär. Mysal üçin,

```
n = n & 0177;
```

n ululykdaky kiçi ýedi bitinden galanlaryny arassalaýar.

| operator (bitler üçin OR) bitleri dikeltmek üçin ulanylýar; ýagny

```
x = x | SET_ON;
```

x ululygyň bitlerinde deňişlilikde SET_ON ululykdaky bar bolan we gabat gelýän 1-lik bitler ýazylýar.

Funksiýalary döretmek we ulanmak. Funksiýany döretmek

Kesgitleme: Funksiýa – bu programmanyň özbaşdak birligi bolup anyk meseläni görkezmek üçin ulanylýar. Si dilinde funksiýa beýleki dillerdäki ýaly podprogrammalary we proseduralary meňzeşdir.

Funksiýany çagyrmak bilen käbir hereketiň ýa-da aňlatmanyň ýerine ýetirilmegine getirýär. Funksiýany döretmäge deňişli bir mysala seredeliň:

```
# include <stdio.h>
# define A “Ashgabat”
# define B “B.Annanow köç.”
# define C “TPI”
```

```
main ( )
{
TPI ( );
Printf (“%s \n”,A);
Printf (“%s \n”,B);
Printf (“%s \n”,C);
TPI ( );
}
TPI ( );
{
int a;
for (a=1; a<=65; a++);
putchar (*);
putchar ('\n');
}
```

Netijede

:

.....
B.Annanow köç.

TPI
.....

Bu programma TPI funksiýany döretdik we ulandyk. Bu funksiýa 65 sany * simwoly ekrana çykarýar.

Funksiýanyň argumenti

Ulanylýan programmanyň netijesinde ýazgyny ortadan ýerleşdireliň. Bu ýagdaýa ulanjak funksiýanyň argumentini görkezmeli bolýarys. Mysal:

```
# include <stdio.h>
# define A "Ashgabat"
# define B "B.Annanow köç."
# define C "TPI"
main ( )
{
TPI ( );
Int b;
KT (29);
Printf ("%s \n",A);
B=(65-len (B))/2;
KT(b);
Printf ("%s \n",B);
KT(b);
Printf ("%s \n",B);
KT((65-len ©)/2);
Printf ("%s \n",C);
TPI ( );
}
TPI ( );
{ int a;
for (a=1; a<=65; a++);
putchar (*');
putchar ('\n');
}
KT©;
Int c;
{
int a;
for (a=1; a<=c; a++);
putchar ( ' ');
}
```

Bu programmada KT()-iň argumentini 3 usulda berildi.

1. KT (29);
2. b=(65-len (b))/2;
KT(b)ž
3. KT ((65-len (c))/2);
KT c
Int c
KT (int c)

Bu programmada A üýtgeýäni 2 ýerde ulandyk. TPI () ulanylýan A bilen KT ulanylýan A biri-biri bilen baglanşyksyz.

Return operator

Bu operator funksiýanyň bahasyny gaýtarmak üçin ulanylýar. Şonuň absolut bahasyny gaýtarmak üçin ulanylýar. Şonuň absolut ululygyny tapmagyň funksiýasyny döredeliň. Goý, bu funksiýa ABŇ bolsun. Funksiýanyň ýerine ýetirilişiniň netijesinde položitel ululyk alynmaly. Bu ululuk hem return operatoryň kömegi bilen gaýtarylýar. Mysal:

```
# include <stdio.h>
main ( )
{
int a=10, b=0, c=-22;
int d, e, f;
d=abs ("%d %d \n", d,e,f);
}
abs(x);
int x;
{
int y;
y = (x<0)?-x;x;
return (y);
}
```

Netijede:

Bu programmada y üýtgeýän ()-ňiki obýektli bolup, onuň bahasyny return operasiýanyň kömegi bilen berilýär.

```
Abs (int x)
{
if (x<0)
return (x);
else
return (x);
}
```

Adresi tapmak, & - operator

& - operator ýerine ýetirilmeginiň netijesinde degişli üýtgeýäniň huşdaky üýtgeýäniň adresi kesgitlenýär.

```
A=24
& a=12126
a=24
printf ("%d %d", a, &a);
Netijede: 24 12126
# include <stdio.h>
main ( )
{
int a=2, b=5;
printf ("main ( )-de a=%d we &a=%d üçin", a&a);
printf ("main ( )-de b=%d we &b=%d üçin ", b&b);
```

```

KT (b);
}
KT (b);
{
printf ("KT ( )-de a=%d we &a=%d üçin", a&a);
printf ("KT ( )-de b=%d we &b=%d üçin", b&b);
}
main ( )-de a=2 we &a=65524
main ( )-de b=5 we &b=65522
KT( )-de a=2 we &a=65514
KT( )-de b=5 we &b=65520

```

Funksiýanyň tipini görkezmek.

Funksiýanyň tipi argumenti bilen bagly dälde, öz bahasy bilen kesgitlenýär. Eger funksiýanyň tipi görkezilmese, onda ol funksiýa int görnüşe degişlidir. Ýönekeý funksiýalara seredeliň.

```
Float KT (x,y);
```

```
Int x,y;
```

```

{
int z;
z=x/y;
return (z);
}

```

Mysal:

S,t hakyky san berlen. Tapmaly:

$f(t, -2s, 1.17) + f(22, t, s-t)$ nirede $f(a, b, c) = 2a - b - \sin c/5 + |c|$

```
# include <stdio.h>
```

```
# include <math.h>
```

```
main ( )
```

```

{
float a, b, c, st, d, e.f();
printf ("s="); scanf ("%f",&s);
printf ("%f",&s);
a=t; b=-2*s; c=1.17, d=KT(a,b,c);
a=2.2; b=t; c=s-t;
d=KT(a,b,c);
printf ("f ( )=%f\n", d+e);
}

```

```
float KT (a,b,c);
```

```
float a,b,c;
```

```

{
float z;
z=(2*a-b-sin c)/(5+tabs (c));
return (z);
}

```


Onlyk sanaýyş ulgamyndaky(s/u) sany ikilik s/u-na geçirýän programma:

```
#include <stdio.h>
#include <conio.h>
int bitcount(unsigned x);
//-----//
main()
{ clrscr();
  unsigned char c;
  while((c = getche()) != 27)
  {
    printf(" --> ");
    bitcount(c);
  }
}
//-----//
/* x sany ikilik sanaýyş sistemasyna geçirýar */
int bitcount(unsigned x)
{
  int b,s[8];
  for(b=1; b <= 8; b++,x >>= 1)
    s[b]=(x & 01);
  for(b=8; b>=1; b--)
    printf("%d",s[b]);

  printf("\n");
  return b;
}
```

Funksiýalar, funksiýanyň täsir edýän çägi we ýadyň klaslary

Huşuň klasy üýtgeýäniň hereketlenen ýaýlasyny we bolmaklygyny dowamlylygyny kesgitleýär. Üýtgeýäniň degişli tipiniň boluşy ýaly degişli husuň klasy hem bolmalydyr.

Huşuň klasy aşakdaky görnüşe eýedir.

№	Huşuň klasy	Ýazylyşy	Dowamlylygy
1.	Awtomatik	awto	wagtlaýyn
2.	Registrlí	register	wagtlaýyn
3.	Statistiki	static	hemişelik
4.	Daşky	extern	hemişelik
5.	daşky statistiki	statik	hemişelik

1. Awto – eger üýtgeýäniň önünde husuň klasy görkezilmese, onda ol husuň awtomatiki klasyna degişlidir. Bu üýtgeýän funksiýanyň içinde gelmelidir. Mysal:

```
# include <stdio.h>
main ( )
```

```

{
awto int a;
...
}

```

2. Registor – registrleri üýtgeýän merkezi prossesoryň registrde ýerleşip, ol huşdaka görä çagyrmaklygy we işi ýerine ýetirmekligi çalt amala aşyrylýar. Köp halatlarda bolsa, huşuň awtomatiki klasy meňzeşdiler. Onuň bolmaklygynyň dowamlylygy wagtlaýyndyr. Mysal:

```

#include <stdio.h>
main ( )
{
register int a;
...
}

```

3. Static – statistiki üýtgeýän awtomatiki üýtgeýäne meňzeşdir, ýöne onuň bolmaklygynyň dowamlylygy hemişelikdir. Huşuň bu klasyna berlen üýtgeýän funksiýanyň içinde berilmelidir. Mysal:

```

#include <stdio.h>
main ( )
{
int a;
for (a=1; a<=3; a++);
{
printf ("Mysal %d", a);
KT ( );
}
}
KT ( );
{
int b=1;
static int c=1;
printf ("b=%d we c=%d \n", b++, c++);
}

```

Netijede:

Mysal 1: b=1 we c=1

Mysal 2: b=2 we c=2

Mysal 3: b=3 we c=3

4. Daşky üýtgeýän extern-huşuň daşky üýtgeýän klasyna degişli bolan üýtgeýän funksiýanyň öňünden berilýär. Bu daşky üýtgeýäniň bolmaklygynyň dowamlylygy hemişelikdir.

```

#include <stdio.h>
int a;
char ch;
main ( )
{

```

```
extern int a;
extern char ch;
}
```

Eger programmada ýeke funksiýa ulanylýan bolsa, onda extern sözünü ýazmaklygyň zerurlygy ýok. Eger birnäçe funksiýa bolsa, onda hökmany görkezilmeli.

5. Daşky statistiki – huşuň statistiki klasyna degişli bolan üýtgeýän hem funksiýanyň önünden berilýär. Daşky üýtgeýän bilen daşky statistiki üýtgeýäniň tapawudy. Daşky üýtgeýäni birnäçe faýlda hem ulanyp bolýar, ýöne daşky statik üýtgeýän diňe bir faýla degişlidir. Mysal:

```
#include <stdio.h>
static int a;
static char ch;
static main ( )
{
    static int a;
    static char ch;
}
```

Si-de esasy ulanylýan huşuň klasy awtomatiki görnüşdir.

Faktorialy hasaplaýan programma:

```
#include <stdio.h>
#include <conio.h>
long fac(long n);
main(void)
{
    clrscr();
    long a=10;
    printf("%ld\n",fac(a));
    getch();
    return 0;
}
//-----//
long fac(long b)
{
    long p=1; long i;
    for(i=1; i<=b; ++i) p=p*i;
    return p;
}
```

```
#include <stdio.h>
#include <conio.h>
void md(int a[], int n);
void main()
```

```

{
    clrscr();
    int i, x[5];
    md(x,4);
    printf("-----\n");
    for(i=0; i<=4; ++i)
        printf("x[%d]=%d\n",i,x[i]);
}
/*-----*/
void md(int a[], int n)
{
    int k;
    for(k=0; k<=n; ++k)
        { printf("a[%d]=",k); scanf("%d",&a[k]); }
}

```

Maglumatlaryň şahalaýyn gurluşy

Şol bir tipe degişli bolan elementleriň toplumyna massiwler diýilýär. Olar aşakdaky ýaly beýan edilýär. Mysal:

```

int a[20]
a[0], a[1], ..., [19]
a[0]=1 a[5]=5
float b[10]
b[3]=5.25
b[6]=2.3

```

Mysal: Islendik 10 sanyň orta bahasyny tapmaklygyň programmasyny düzmeli.

```

#include <stdio.h>
#define num 10
main ( )
{
    int i, a, s, c[num];
    for (i=0; i<num; i++);
    scanf ("%d", & c[i]);
    for (i=0; i<num; i++);
    printf ("%d", c[i]);
    printf ("\n");
    for (s=0; i=0; i<num; i++);
    s+=c[i];
    a=s/num;
    printf ("orta baha = %d", a);
}

```

Massiwler huşun degişli klasyna degişli bolup durýar. Mysal:

```

#include <stdio.h>
int a[20];
main ( )

```

```

{
float b[30];
static char ch[10];
extern int a[ ];

```

Mysal: Her aýyň degişli günleri ekrana çykarmagyň programmasyny düzeliň.

```

#include <stdio.h>
int gun[12]={ 31, 29, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 }
main ( )
{
int a;
extern int gun [ ];
for (a=0; a<12; a++);
printf ("%d-ay, %d-gun \n", a=1, gun[a]);

```

Görkeziji operasiýasy.

Görkeziji operasiýasy üýtgeýäniň simwoliki bahasyny görkezýär. * görkeziji operasiýasy. Mysal:

```

A=2;
B=&a;
C=*b;
C=2;
#include <stdio.h>
main ( )
{
int a[4]*b, c;
float a[4]*a;
b=a; (b=2a[0]);
e=d; (e=&d[0]);
for (c=0; c<4; c+1)
printf ("görkeziji %d u we %d u", c, b+c, e+c);
}

```

Netijede: görkeziji: 0; 65518 we 65498

1; 65580 we 65502

2; 65522 we 65506

3; 65524 we 65510

Mysal1: Matrisalaryň jemini tapar ýaly programma düzmeli.

1 2 5 1 6 3

 + =

3 4 0 2 3 6

```
# include <stdio.h>
```

```
# include <math.h>
```

```
main ( )
```

```
{
```

```
int i, j, c[2][2];
```

```

static int a[2][2]={ 1,2,3,4};
static int b[2][2]={ 5,1,0,2};
for (i=0; i<2; i++); { for (j=0; j<2; j++)
c[i][j]=a[i][j]+b[i][j];}
for (i=0; i<2; i++) { for (j=0; j<2; j++)
printf (“\n c[%d, %d]=%d”, i+1, j+1, c[i][j]);
}}

```

Mysal2: Matrisalaryň köpeltmek hasyly alynar ýaly programma düzmeli.

```

1 2    5 1    5 5

```

```

*      =

```

```

3 4    0 2    15 11

```

```

#include <stdio.h>

```

```

#include <math.h>

```

```

main ( )

```

```

{

```

```

int i, j, c[2][2];

```

```

static int a[2][2]={ 1,2,3,4};

```

```

static int b[2][2]={ 5,1,0,2};

```

```

for (i=0; i<2; i++); { for (j=0; j<2; j++){

```

```

c[i][j]=0;

```

```

for (k=0; k<2; k++)

```

```

c[i][j]=a[i][k]*b[k][j];} }

```

```

for (i=0; i<1; i++) { for (j=0; j<2; j++)

```

```

printf (“\n g[%d, %d]=%d”, i+1, j+1, c[i][j]);

```

```

}}

```

```

#include <stdio.h>

```

```

#include <conio.h>

```

```

#include <stdlib.h>

```

```

void mg(int d);

```

```

void main()

```

```

{

```

```

clrscr(); randomize();

```

```

int i;

```

```

extern float x[10];

```

```

mg(10);

```

```

for(i=0; i<10; ++i)

```

```

printf("a[%d]=%3.1f\n",i,x[i]);

```

```

getch();

```

```

}

```

```

/*-----*/

```

```

float x[10];

```

```

void mg(int m)

```

```

{

```

```

int i;
for(i=0; i<m; ++i)
    x[i]=random(100)*0.2;
}

```

Simwol setirli char tipe degişli massiw bolup soňky ‘10’ simwol bilen gutarýar. Simwol setiri girizilende köp halatda huşuň ölçegi anyk görkezilýär. Mysal üçin: char a[20] = “bahar geldi”. Bu ýagdaýda huşda 20 öýjük goýulýar we onuň 12-si simwollar bilen doldurylýar. Galanlary bolsa, ýagny galan 8 öýjük ‘10’ simwoly bilen doldurylýar. Simwol setiri hökmany ýagdaýda huşuň statistiki ýa-da daşky klasyna degişli bolmalydyr.

Eger static char a[] = { ‘B’, ‘a’, ‘h’, ‘a’, ‘r’, ‘ ‘, ‘g’, ‘e’, ‘l’, ‘d’, ‘i’, ‘.’, ‘%’ }

Eger setir döretjek bolsa, onda * operasiýasyny ulanmaly. Mysal üçin: static char *a= “Bahar geldi”. Bu ýagdaýda a massiwiň bir elementi bar, ol hem bir setirden ybarat. Eger a massiwiň simwoly birden köp bolsa, onda ol görkezilýär. Mysal üçin: static char *a[3]={ {salam}, {sag bol}, {bahar geldi} }; Bu ýerde a [0] elementde “salam”, a[1] elemente “sag bol”, a[2] elementde “bahar geldi”, degişlidir. Setirleriň ölçegini anyk görkezip hem bolýar. Mysal üçin: 1.static char a[3][8]

2. static char *a[3]

Bularyň tapawudy birinji ýagdaýda her setirdäki simwollara 8 öýjik goýulýar, ýöne ikinji ýagdaýda gerekli ýer goýulýar. Ýagny ikinji ýagdaýda artykmaç ýer eýelenmeýär. Setirleri girizmek we çykarmak üçin degişlilikde gets() we puts() funksiýalary ulanylýar.

1. gets () funksiýasy – bu funksiýa setiri girizmek üçin ulanylýar. Ol iki ädime amala aşyrylýar.

1. huşda ýer bellemek

2. funksiýa ulanmak

Huşda ýer bellemek bolanda giriziljek setiriň ölçegini bermekdir. Mysal üçin char a[81] funksiýa ulanmakda esasan gets() funksiýasy ulanylýar. Bir mysala seredeliň:

```

#include <stdio.h>
main ( )
{
    static char a[81];
    printf (“Adyň näme?\n”);

    gets (a);
    printf (“%s gowy ad\n”, a);
}

```

Netijede:

Adyň näme?

Myrat

Myrat gowy ad

Gets () funksiýa bilen scanf funksiýanyň tapawudy scanf () funksiýasy % format bilen söz okaýar. Ýöne gets () bolsa, tutuş setiri okaýar.

2. Puts funksiýasy – kabul edilen setiri çapa çykarýar. Argumentinde çapa çykmalý setiri görkezmeli. Mysal:

```
# include <stdio.h>
main ( )
{
static char a[ ]= “bahar geldi”;
static char *b= “gyş geldi”;
puts (“ýagyş ýagýar”);
puts (a);
puts (b);}
Netijede:  ýagyş ýagýar
           bahar geldi
           gyş geldi
```

printf () bilen puts () funksiýanyň tapawudy: puts () funksiýasy her setiri awtomatik täze setirden ýazýar. Ýöne printf ()-de bolsa ‘\n’ simwoly ýazmaly bolýarys. Setirler bilen esasan dört sany funksiýa işleýär: strlen (), strcat (), strcmp (), strcpy ().

1. strlen () – bu funksiýa setiriň uzynlygyny kesgitleýär. Mysal:

```
# include <stdio.h>
main ( )
{
static char *a = “bahar geldi”;
printf (“%d \n”, strlen (a));
}
```

2. strcat () – bu funksiýa iki setiri birleşdirmek üçin ulanylýar. Mysal:

```
# include <stdio.h>
main ( )
{
static char *b = “ýagdaýlaň gowymy?”;
puts (“adyň näme?”);
gets (a);
strcat (a,b);
puts (a);
}
```

Netijede: Adyň näme? Bahar. Bahar ýagdaýlaň gowymy?

3. strcmp () – bu funksiýa setirleri deňeşdirmek üçin ulanylýar. Mysal:

```
# include <stdio.h>
# define A “Bahar”
main ( )
{
static char a[40]
puts (“Adyň näme?”);
gets (a);
while (strcmp (a, A)!=0);
{
```



```
puts ("ýalňyş");
gets (a);
}
puts ("dogry");
}
```

4. strcpy () – bu funksiýasy setiriň kopiýasyny almak üçin ulanylýar. Mysal:

```
# include <stdio.h>
# define A "Ýalňyş"
main ( )
```

```
{
static char *a=A;
static char b[40]
puts (a);
puts (b);
strcpy (b, a);
puts (a);
puts (b);
Netijede: ýalňyş barla
         ýalňyş barla
         ýalňyş barla
```

Turbo c++-de giriş çykyş

Si diliniň kitaphanasynda üç ýol bilen girip bolýar.

1. Awtomatiki
 2. Faýla geçmek ýoly (# include)
 3. Kitaphana giriş (lc.lib)
- Kitaphanadan biziň ulanan funksiýalarymyz:

1. Giriş-çykyş funksiýalary
 - getchar() – simwol almak
 - putchar() – simwol çap etmek
 - gets () – setiri almak
 - puts () – setiri çap etmek
 - scanf – formatirlenen giriş
 - printf – formatirlenen çykyş
2. Setir bilen işleýän funksiýalar.
 - Strlen () – setiriň uzynlygy
 - Strcat () – setirleri birleşdirmek
 - Strcmp () – setirleri deňeşdirmek
 - Strcpy () – setirleriň kopiýasyny almak
3. Faýyllary açmak – ýapmak unksiýasy.
 - Fopen () – faýly açmak
 - Fclose () – faýly ýapmak
 - Gets () – faýyldan simwol almak
 - Puts () – faýlda simwoly çap etmek

1) Fopen () funksiýasy – bu funksiýa faýly açmak üçin ulanylýar. Fopen () funksiýasy üç sany esasy parametri dolandyrýar:

a) Faýlyň ady

b) Faýly nähili ulanjak:

- “r” – okamak
- “w” – ýazmak
- “a” – goşmak

Bellik: eger bar bolan faýla “w” ulansak, onda öňki ýazgylar ýitip, täze ýazgylar galýar.

c) Faýly görkeziji bolup durýar.

FILE in:

In = fopen (“ady”, “r”);

Bu ýerde in “ady” diýlen faýla görkeziji bolup durýar.

2) Fclose () – bu funksiýa faýly ýapmak üçin ulanylýar. Fclose (in); ulanylýar. Bu ýerde argumentde faýlyň görkezijisi ýazylmaly.

3) Gets () – Puts () faýyllary girizmek we çykarmak.

Ch=gets (in); - bu görkezijidäki faýldan simwol almak.

Puts (ch, stdout); – bu ch simwoly faýla ýazmak.

Puts (ch, stdout); - simwoly standart çykyşa ibermek. Mysal:

```
# include <stdio.h>
```

```
main ( )
```

```
{
```

```
FILE *in;
```

```
Int ch;
```

```
If ((in=fopen (“ady”, “r”))!=NULL)
```

```
{
```

```
while ((ch=gets (in)!=EOF)
```

```
puts (ch, stdout);
```

```
fclose (in);
```

```
}
```

```
else
```

```
printf (“faýl açylmaly \n”);}
```

mesele:

n sany otag bar. Her otagyň gapysynda onuň tertip boýunça ýazylan(N1, N2, ..., Nn). Otaglarda çyra bar. Ol çyra şol otagdaky açaryň kömegi bilen ýakylýar ýa-da öçürilýär. Her otagdaky açaryň iki sany durnukly ýagdaýy bar: 0,1; 0 ýagdaýda otagdaky çyra öçýär, 1 ýagdaýda bolsa otagdaky çyra ýanýar. Açaryň ýagdaýy onuň düwmejigine basylmak bilen üýtgedilýär. Başlangyç ýagdaýda hemme otaglardaky çyralar öçürilen. Ilkinji okuwçy hemme otaglara girip açarlary basyp çykýar. Soňra bolsa ikikji okuwçy her ikinji otaga girip, üçünji okuwçy bolsa her bir üçünji otaga girip we ş.m., n-nji okuwçu bolsa diňe n-nji otagdaky açaryň ýagdaýyny üýtgedip gaýdýar. Iň soňunda näçe otagda çyra ýanyk galar? Bu meseläni çözmek üçin programma ýazmaly.

```

#include <stdio.h>
#include <conio.h>
#include <iostream.h>
main()
{
    clrscr();
    int a[1001],i,j,n,s=0;
    cout<<"n="; cin>>n;
    for(i=1; i<=n; i++) a[i]=0;

    for(i=1; i<=n; i++)
    for(j=i; j<=n; j=j+i)
    if(a[j]==0) a[j]=1; else a[j]=0;

    for(i=1; i<=n; i++)
    if(a[i]>0) s++;

    printf("s=%d",s);
    getch();
}

```

mesele:

Temperaturanyň ferengeýtdäki bahasyny selsä geçirmek üçin programma ýazmaly

```

#include <stdio.h>
#include <conio.h>
main()
{
    clrscr();

    float fahr, celsius;
    int lower, upper, step;
    lower=0;
    upper=300;
    step=20;

    celsius=lower;
    printf("Celsius \t Fahr\n");
    printf("-----\n");
    while( celsius <= upper )
    {
        fahr=(9.0)/(5.0)*celsius+32;
        printf("%6.0f\t%3.1f\n",celsius,fahr);
        celsius = celsius + step;
    }
}

```

```

    getch();
}

```

Turbo C++ dili öz ülanýjylaryna diňe bir programma ýazmaklyga däl-de eýsem 300-den gowrak biblioteka functionlary hem-de macrolary ulanmaklyga mümkinçilik berýär. Şeýle macrolaryň we functionlaryň köpüsi C diliniň dürli görnüşleri bilen ylalaşykly ulanylyp biliner.

Turbo C++ diliniň hemme bibliotekalary üçin interface faýly döredilendir(.h giňeldilmeli faýllar) döredilendir. Ol faýllarda hemme biblioteka functionlarynyň prototipleri saklanýarlar.

Mesele:

girizilen tekstde näçe sany setiriň, boşlugyň, sifriň bardygyny hasaplaýan programma ýazmaly:

```

#include <stdio.h>
#include <conio.h>
int main(void)
{
    clrscr();
    int c, i, nwhite, nother;
    int ndigit[10];
    nwhite = nother = 0;
    for (i=0; i<10; ++i) ndigit[i]=0;
    while((c=getch()) != 27)
        if (c>='0' && c<='9') ++ndigit[c-'0'];
        else if (c==' ' || c=='\t') ++nwhite;
        else ++nother;
    printf("numbers=");
    for (i=0; i<10; ++i) printf(" %d",ndigit[i]);
    printf("\nspace=%d other=%d\n",nwhite, nother);
    getch();
    return 0;
}

```

mesele:

Bitin sany bitin derejä götermäge mümkinçilik berýän programma ýazmaly

```

#include <stdio.h>
#include <conio.h>
int power(int m, int n);
main(void)
{
    clrscr();
    int i;
    for(i=0; i<10; ++i)
        printf("%d\t%d\t%d\n",i,power(2,i),power(-3,i));
    getch();
}

```

```

    return 0;
}
int power(int base, int n)
{
    int i,p;
    p=1;
    for(i=1; i<=n; ++i)
        p=p*base;
    return p;
}

```

mesele:

Girizilen setiriň içinden iň uzynyny kesgitlemäge mümkinçilik berýän programma:

```

#include <stdio.h>
#include <conio.h>
#define MAXLINE 1000
int getline(char line[], int maxline);
void copy(char to[], char from[]);
main(void)
{
    int len;
    int max;
    char line[MAXLINE];
    char longest[MAXLINE];
    clrscr();
    max=0;
    while ((len=getline(line, MAXLINE)) > 0)
        if (len > max)
        {
            //      printf("%s --> %d\n",line,len);
            max=len;
            copy(longest, line);
        }
    if(max>0) printf("%s\n",longest);

    getch();
    return 0;
}
//-----//
int getline(char s[], int lim)
{
    int c,i;
    for(i=0; i<lim-1 && (c=getche()) != 13; ++i)
        s[i]=c;
}

```

```

if (c=='\n')
{
    s[i]=c; ++i;
}
s[i]='\0';
printf("\n");
return i;
}
//-----//
void copy(char to[], char from[])
{
    int i;
    i=0;
    while((to[i] = from[i]) != '\0')
        ++i;
}

```

Turbo C++ diliniň ýörite serişdeleri

C diliniň preprocessor kompilyatora berilmezden öň, tipi *.cpp bolan, tekst faýllary işlýär. Preprocessor hemme macroçagyryşlary giňeltýär we daşky faýllary ýerine goýýar.

Preprocessorň komandalary

Preprocessorň komandasyny aňlatmak üçin # belgi ulanylýar. Bu simwili (Turbo) C dilinde başga maksat üçin ulanylmaýar. Preprocessorň komandalarynyň sanawy:

komanda	beýany
#define	makrony kesgitlemek
#undef	Makronyň kesgitlemesini inkär etmek
#include	daşky faýldan tekst goýmak
#if	aňlatma hemişeligiňe baglylykda teksti şertli goýmak
#ifdef	Makronyň kesgitlemegine baglylykda tekstiň bölegini goýmak
#ifndef	makro kesgitlemedik bolsa tekst bölegini goýmak
#else	#ifdef we #ifndef üçin alternatiwa
#endif	şertli goýulýan tekst böleginiň soňyny kesgitlemek
#iline	setiriň nomerini görkezýär
#defined	#if bilen bilelikde ulanylýar. Görkezilen adyň makronyň adydygyny ýa-da dældigini kesgitleýär.
#pragma	kompileýator ýerine ýetýän döwründe görkezme berýär
#error	kompileýasiýa döwründe ýüze çykýan ýalňyşlyklar barada habary ýüze çykarýar

Turbo C++ dilinde C diliniň täze ANSI standarty kesgitlenendir. Diliň bu

görnüşünde köp täzelikler girizilendir. Umuman programmirleme dilleri aşak derejeli hem-de ýokary derejeli diýilip atlandyrylýan iki topara bölünýärler. C dili bolsa olaryň ikisine hem degişli bolan ortalık bir dil bolup onuň mümkinçilikleri juda uludyr.

maglumatlaryň tipleri:

Tipler	Uzynlygy	Çägi
=====+=====+=====		
==		
unsigned char	8 bits	0 to 255
char	8 bits	-128 to 127
enum	16 bits	-32,768 to 32,767
unsigned int	16 bits	0 to 65,535
short int	16 bits	-32,768 to 32,767
int	16 bits	-32,768 to 32,767
unsigned long	32 bits	0 to 4,294,967,295
long	32 bits	-2,147,483,648 to 2,147,483,647
float	32 bits	$3.4 * (10^{**-38})$ to $3.4 * (10^{**+38})$
double	64 bits	$1.7 * (10^{**-308})$ to $1.7 * (10^{**+308})$
long double	80 bits	$3.4 * (10^{**-4932})$ to $1.1 * (10^{**+4932})$

sys_errlist massiwiniň elementleri we olaryň aňladýan manylary:

Elemanti	Manysy
=====+=====	
E2BIG	Arg list too long
EACCES	Permission denied
EBADF	Bad file number
ECONTR	Memory blocks destroyed
ECURDIR	Attempt to remove CurDir
EDOM	Domain error
EEXIST	File already exists
EFAULT	Unknown error
EINVACC	Invalid access code
EINVAL	Invalid argument
EINVDAT	Invalid data
EINVDRV	Invalid drive specified
EINVENV	Invalid environment
EINVFMF	Invalid format
EINVFNC	Invalid function number
EINVMEM	Invalid memory block address
EMFILE	Too many open files

ENMFILE	No more files
ENODEV	No such device
ENOENT	No such file or directory
ENOEXEC	Exec format error
ENOFILE	No such file or directory
ENOMEM	Not enough memory
ENOPATH	Path not found
ENOTSAM	Not same device
ERANGE	Result out of range
EXDEV	Cross-device link
EZERO	Error 0

Edebiýatlar

1. Türkmenistanyň Konstitusiyasy. Aşgabat, 2008.
2. Gurbanguly Berdimuhamedow. Ösüşin täze belentliklerine tarap. Saýlanan eserler. I tom. Aşgabat, 2008.
3. Gurbanguly Berdimuhamedow. Ösüşin täze belentliklerine tarap. Saýlanan eserler. II tom. Aşgabat, 2009.
4. Gurbanguly Berdimuhamedow. Garaşsyzlyga guwanmak, Watany, halky söýmek bagtdyr. Aşgabat, 2007.
5. Gurbanguly Berdimuhamedow. Türkmenistan – sagdynlygyň we ruhubelentligiň ýurdy. Aşgabat, 2007.
6. Türkmenistanyň Prezidenti Gurbanguly Berdimuhamedowyň Ministrler Kabinetiniň göçme mejlisinde sözlän sözi. (2009-njy ýylyň 12-nji iýuny). Aşgabat, 2009.
7. Türkmenistanyň Prezidentiniň “Obalaryň, şäherleriň, etrapdaky şäherçeleriň we etrap merkezleriniň ilatynyň durmuş-ýaşayyş şertlerini özgertmek boýunça 2020-nji ýyla çenli döwür üçin” Milli maksatnamasy, Aşgabat, 2007.
8. “Türkmenistany ykdysady, syýasy we medeni taýdan ösdürmegiň 2020-nji ýyla çenli döwür üçin Baş ugry” Milli Maksatnamasy, “Türkmenistan” gazetini, 2003-nji ýylyň 27-nji awgusty.
9. “Türkmenistanyň nebitgaz senagatyny ösdürmegiň 2030-njy ýyla çenli döwür üçin Maksatnamasy”. Aşgabat, 2006.
10. The C Programming language, Second Edition, Brain W.Kernigan, Deniss M. Ritchie. Colorado Uniwersiteti, 1982.
11. Turbo C at Any Speed, Richard S. Wiener, Uniwersity of Colorado at Colorado Spring Colorado Uniwersiteti, 1988.
12. Б. Керниган, Д.Ритчи Язык программирования С++, Москва “Финансы и статистика” – 1992.
13. Р.Уинер. Язык ТУРБО С, Москва, 1991.
14. Дэвид Дж.Круглинский, Скотт Уингоу, Джордж Шеферд (Для профессионалов) Программирование на Microsoft VISUAL С++ 6.0, Санкт-Петербург, Москва, Харьков, Минск, 2002.
15. Н.Культин С/С++ в задачах и примерах, Санкт-Петербург “БХВ-Петербург”, 2008.

MAZMUNY

Giriş	1
Programma ýazmagyň düzgüni	3
Turbo c++ diliniň mümkinçiliklerine syn	6
Üýtgeýän ululyk we arifmetiki aňlatma	9
for operatory	15
Atlandyrylan konstantalar(hemişelikler)	16
Literany çapa çykarmak	17
Faýlyň kopiýasyny almak	17
Literi sanamak	19
Setiri sanamak	21
Söz sanamak	22
Massiwler	23
Funksiýalar	26
Argumentler. Bahasy boýunça çagyrmak	29
Literleriň massiwi	30
Daşky üýtgeýän ululyklar we olaryň täsir edýän çägi	33
Tipler, operatorlar we aňlatmalar	37
Üýtgeýän ululyklaryň atlary	37
Maglumatlaryň tipleri we ölçegleri	38
Hemişelikler	39
Deklarirlemek	42
Arifmetiki operatorlar	43
Gatnaşyk operatorlary we logiki operatorlar	43
Tipleri özgertmek	44

Inkrement we dekrement operatorlar	48
Bitleýin operatorlar	50
Funksiýalary döretmek we ulanmak. Funksiýany döretmek	51
Funksiýanyň argumenti	52
Return operator	53
Adresi tapmak, & - operator	53
Funksiýalar, funksiýanyň täsir edýän çägi we ýadyň klaslary	55
Maglumatlaryň şahalaýyn gurluşy	58
Turbo c++-de giriş çykyş	63
Turbo C++ diliniň ýörite serişdeleri	68
Edebiýatlar	70