

Konceptet Bazë të programimit në gjuhën C

4.1 Elementët bazë të gjuhës C

Ne 1972, ne Laboratorin Bell, Dennis Ritchie projektoi dhe realizoi versionin e parë të gjuhës C. Ritchie ka zhvilluar disa nga principet bazë të sintaksës të gjuhës BCPL, krijuar nga Martin Richards, dhe të gjuhës B, krijuar nga Ken Thompson, autor i sistemit operativ Unix. Me pas Ritchie dhe Thompson rishkruajtën në C të gjithë sistemin operativ të Unix.

Ne 1983 Instituti Kombëtar Amerikan për Standart-et (ANSI) filloi standartizimin e **gjuhës C**, e cila përfundoi në 1989 me përcaktimin ANSI C, e cila më vonë është përditësuar në përputhje me filozofinë e programimit object-oriented, duke sjellë gjuhën C ++, (file me prapashtesë . CPP).

Një gjuhë dhe në mënyrë të veçantë një gjuhë programimi, karakterizohet nga një *alfabet*, nëpërmjet të cilit bëhet i mundur ndërtimi i fjalëve duke përdorur rregullat *leksikografike*. Fjalët e një gjuhe njihen nga fakti që ato i binden rregullave të saj dhe përfshihen ndërmjet ndarësve. Një numër karakteresh mund të marrin një kuptim të caktuar në gjuhë: këta quhen *operatorë*.

Ekzistojnë fjalë me një kuptim të veçantë që quhen *fjalë çelës* ose *fjalë të rezervuara*. Bashkësia e fjalëve, ndarësve dhe fjalëve çelës përbëjnë *frazat* e gjuhës, të quajtura gjithashtu *instruksione* në rastin e gjuhëve të programimit.

Ndërtimi i frazave i bindet rregullave *sintaksore*, të cilat përbëjnë dhe *gramatikën* e gjuhës. Gramatika e një gjuhe programimi është e veçantë pasi ajo nuk lejon dykuptueshmëri si në rastin e gjuhëve të zakonshme. Kjo lehtëson dallimin ndërmjet një fraze korrekte dhe një fraze jo korrekte.

Gjuha C lejon menaxhimin e bit-eve, byte-ve dhe adresave të kujtesës, në dallim nga gjuhët e tjera të nivelit të lartë si **Basic** dhe **Pascal**, për këtë arsye ndonjëherë "quhet një gjuhë e nivelit mesatar". Gjithashtu ajo ka vetëm 32 fjalë kyçe, shumë pak në krahasim me Basic, e cila ka me shume se 150.

Pergatitja dhe ekzekutimi i nje programi

Zhvillimi i nje programi perbehet nga 3 faza:

- 1. Shkrimi i kodit:** kjo faze konsiston ne shkrimin e kodit te programit ne nje gjuhe specifike programimi duke perdorur nje perpunues teksti. Programi I shkruar quhet file burim (.c)
- 2. Kompilimi:** ne kete faze verifikohet saktesia ose korrektesa nga ana gramatikore, sintaksa, kompatibiliteti midis tipeve te variablave etj. te programit. Ne vazhdim file burim perkthehet ne file te ekzekutueshem(.exe).
- 3. Ekzekutimi:** Pasi behet gati *file exe*, atehere ekzekutojme programin me komanden perkatese. Per verifikimin e saktetise te kodit te shkruar, programet mund te ekzekutohen edhe ne menyren hap pas hapi.

Kompilatori i C kërkon një listë të shkruar në text format, të cilat ju mund ta shkruani me një editor cfaredo teksti. File tekst i programit, i cili zakonisht ka prapashtesë **“.C”** quhet "Program burim" në fillim kompilohet, dmth përkthehet dhe transformohet në një file të ri të quajtur "Program

Bazat e Informatikës, Programimi në gjuhën C

objekt” dhe për ti dalluar “nga njëri-tjetri, programi i përpunuar që të bëhet i kuptueshem vetëm në gjuhë makine merr një emër të ri me prapashtesë “.OBJ”.

Te gjithë sistemet e **Gjuhes C** në përgjithësi përbëhet nga tri pjesë: **Mjedisi; Gjuha dhe Biblioteka Standarte të C.**

Programet e shkruara në C zakonisht kalojnë nëpërmjet 6 faza, përpara se të ekzekutohen. Këto janë:

1. **Editimi**
2. **Perpunimi**
3. **Compilimi**
4. **Lidhja**
5. **Ngarkimi**
6. **Ekzekutimi**

Hapi i parë është që të shkruani kodin (redaktimin) në një skedar: kjo është bërë me një “Program të quajtur Editor”. Programuesi shkruan programin në gjuhën C nëpërmjet editorit të saj dhe më pas kryen korrigjimet. Programin e ruajme në memorie disk dhe emri i file të programit të shkruar duhet të përfundojë me prapashtesë **.C**.

Programuesi më pas kompilon (redakton) programin. Kompilatori e përkthen kodin e programit C në gjuhë makine (i quajtur ndryshe edhe kod objekti). Sistemi në gjuhën C para se të fillojë procesin e përkthimit, programi automatikisht do të kandidojë në preprocessor.

Preprocessor i gjuhes C i bindet komandave të veçantë të quajtur direktiva preprocessor, me të cilin ajo është deklaruar dhe programi duhet të kryejë manipulime të caktuara, para kompilimit aktual. Këto manipulime kryesisht konsistojnë në përfshirjen e dosjeve të tjera që të plotësohen, dhe zëvendëson simbole të veçanta në një tekst të programit.

Faza e katërt është quajtur lidhja (linkimi). Programet në C zakonisht përmbajnë referenca të funksioneve të përcaktuara gjetiu, psh në bibliotekat standarde apo ato të një grupi i programuesve duke punuar në një projekt të veçantë.

Lidhja midis kodit objekt me atë të funksioneve të humbur prodhon një imazh ekzekutimi.

Faza e pestë është quajtur ngarkimi. Para se të ekzekutohet një program duhet të ngarkohet në kujtesë. Ky veprim kryhet nga **Ngarkuesi** (loader), i cili merr imazhin e ekzekutueshem nga disku dhe e transferon në kujtesë. Së fundi, nën kontrollin e CPU saj, kompjuteri do të ekzekutojë programin.

Më poshtë po përshkruajmë: ***Elementët e gjuhës së programimit C.***

Alfabeti i gjuhës C përfshin Gërmat e mëdha: nga A në Z (26 karaktere); Gërmat e vogla: nga a në z (26 karaktere); Karakterin bosh (hapesirë): (1 karakter); Shifrat dhjetore: nga 0 në 9 (10 karaktere); Simbole speciale; operatorë, shënja pikësimi dhe ndarëset.

Kur shkruajme kodin është mirë të përdorim komente dhe për të komentuar me shumë se një rresht duke përdorur simbolet:

- ⇒ **/* për të hapur komentin */ për ta mbyllur**
- ⇒ **// për të komentuar një rresht të vetëm**
- ⇒ **Çdo veprim mbaron me ;(pike presje)**

4.2 Paraqitja e një programi në C

Gjuha e programimit C

Siç kemi thënë, algoritmi përmban veprimet që duhet të kryhen për zgjidhjen e një problemi. Një program përfshin jo vetëm algoritmin “të përkthyer” sipas rregullave të gjuhës, por ai duhet të kapë dhe të regjistrojë informacionin e dhënë, të bëjë llogaritje dhe regjistrime të rezultateve të ndërmjetme dhe së fundi të afishojë rezultatet. Shohim me poshte nje kod programi qe afishon “Miredita” ne ekran:

```
#include <stdio.h>
main( )
{
    printf("Miredita!\n");
}
```

Ky program përfshin biblioteken standarte stdio.h, e cila është një skedar tekst me strukture dhe makro të percaktuar. Keto skedare quhen ndryshe koke (header) dhe mbarojne me .h.

Percaktojme funksionin main, i cili nuk merr asnjë vlerë si vlerëdhënie. Programi fillon ekzekutimin duke u nisur nga ky funksion i cili duhet të jetë prezent në çdo program.

Struktura e një program

1. Deklarimi i librarive
2. Deklarimi i konstanteve
3. Funksioni main()
4. Kllapa gjarpëruese hapese {
5. Deklarimi i variablave
6. Inicializimi ose dhënie e vlerave fillestare
7. Zgjidhja e problemit
8. Afishimi i rezultatit
9. Mbyllja e programit me kllape gjarpëruese mbyltese }

Deklarimi i librarive

- #include **stdio.h** - standard library I/O file scanf() dhe printf()
- #include **math.h** Libraria standarte e fx matematikore
(sin(x); - sinh(x); cos(x); - cosh(x); tan(x); - tanh(x); exp(x); pow(x, y); fabs(x); - abs(x);

sqrt(x); log(x); - log10(x);)
- #include **ctype.h**
- #include **string.h**

Fillojme duke ekzaminuar programin si me poshte:

```
#include<stdio.h>
main()
{
    printf("abcçd");
    printf("dheëf");
    printf("ggjhi");
    printf("jklilm");
    printf("nnjopq");
    printf("rrrssht");
    printf("thuvx");
    printf("xhyzzh");
}
```

Duke e ekzekutuar paraqiten në ekran germat e alfabetit shqiptar:

Bazat e Informatikës, Programimi në gjuhën C

abcçddheëffggjhijklmllmnnjopqrrrsstthuvxxhyzzh

Duke filluar nga fjala main dhe gjithcka ndodhet midis kllapat gjarperushe {} kemi ekzekutimin e programit. Trupi i programit është i përbërë nga një seri instruksionesh dalje printf që ekzekutohen në mënyrë lineare dhe një herë të vetme.

Istruksioni printf afishon në ekran textin që ndodhet brenda kllapave () dhe "" . Psh:

```
printf("abc"); afishon abc
```

Nëse duam që rezultati pas çdo instruksion printf të jetë në një rresht të ri duhet të përdorim simbolin \n në fund të çdo stringu para mbylljes se thonjezave". Psh:

```
#include <stdio.h>
main()
{
    printf("abc\n");
    printf("def\n");
    printf("ghi\n");
}
```

Duke ekzekutuar programin në ekran do të afishoheshin stringjet e karaktereve të mëposhtem:

```
abc
def
ghi
```

Në këtë rast stringu i parë (abc) afishohet në ekran në pozicionin aktual të kursorit. Nëse duam që të fillojë afishimi në një rresht të ri mjafton të vendosni \n edhe në fillim si psh:

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

Ketu fillimisht kalohet në një rresht të ri, pastaj afishohet stringu që duam dhe kursori pozicionohet në një rresht të ri. Sekuenca \n i korrespondon një karakteri të fjalës (newline) rresht i ri.

Identifikatorët

Identifikatorët janë emrat që ju japin programeve, konstanteve, variablave dhe pjesëve të tjera të një programi. Identifikatorët duhet të fillojnë detyrimisht me një gërmë (dhe jo me simbole të tjera) të pasuar nga një numer çfarëdo (pakufizim) gërmash ose shifrash.

P.sh. : Shpejtësia X1 Shuma Epsilon janë identifikatorë korrekt, ndërsa :
1X +M !Shuma nuk janë korrekt sepse nuk fillojnë me gërmë.

Identifikatorët zakonisht i zgjedhim sinjifikativë për të shprehur sa më qartë atë që ata paraqesin. P.sh. për shpejtësinë fillestare të një trupi zgjedhim V0, për rrënjët e një ekuacioni zgjedhim X1, X2, etj.

Fjalët çelës të C se nuk duhet të përdoren si identifikatorë meqë ato kanë një kuptim special në gjuhë. Gjuha C përmban edhe disa identifikatorë standartë siç janë: int, real, true, scanf, printf, etj. Këta identifikatorë kanë një kuptim të paracaktuar, por nuk janë fjalë çelës. Si rrjedhojë, ata mund të ricaktohen prej programuesit, por një gjë e tillë nuk është e këshillueshme.

Gjuha C është shumë i ndjeshëm në rastin e përdorimit të gërmave të mëdha dhe të vogla.

Psh emrat e variablave: NUMER, Numer ose NumeR janë të ndryshme.

Fjalet celes si printf() duhet të shkruhen në germa të vogla.

Rregull atëherë do të jetë për të shkruar variabla me germa të vogla, përdorim “_”:

```
int numer_1;
```

```
int prova1_prova;
```

Konstantet

Janë madhësi të cilat nuk e ndryshojnë vlerën e tyre gjatë ekzekutimit të programit. Ato mund të jenë numerike, karakter ose llogjike. Një konstante int (numer i plotë) është një zinxhir shifrash të cilat nuk duhet të përfshijnë presjen ose pikën dhjetore; konstantet int negative duhet të paraprihen nga shenja minus. Kështu :

0 137 -2516 +17745

janë konstante int të vlefshme ndërsa ato që vijojnë janë të pavlefshme :

5,280 (Presjet nuk lejohen në konstantet e plota)

16.0 (Konstantet e plota nuk duhet të përmbajnë pikë dhjetore)

--5 (Vetëm një shenjë algjebrike është e lejuar)

7- (Shenja algjebrike duhet t'i paraprijë vargut të shifrave)

Një konstante e tipit float dhe double mund të paraqitet si numër i zakonshëm dhjetor ose në formë eksponenciale.

Paraqitja dhjetore e konstanteve reale duhet të përfshijë ekzaktesisht një pikë dhjetore (jo presje), por nuk duhet të fillojë ose të përfundojë me pikë dhjetore. Për numrat me shenjë vlejné të njejtat rregulla si më sipër. Kështu :

1.234 -0.1536 +56473.0

janë konstante reale të vlefshme, ndërsa ato që vijojnë janë të pavlefshme :

1,75263 (presja nuk lejohet në konstantet reale)

82 (konstantet reale në formën dhjetore duhet të përmbajnë pikë dhjetore)

.01 (konstantet reale nuk duhet të fillojnë me pikë dhjetore)

24. (konstantet reale nuk duhet të përfundojnë me pikë dhjetore)

Paraqitja eksponenciale ose me presje notuese konsiston në një konstante të plote ose reale në formë dhjetore e ndjekur nga germa E dhe një konstante të plote që interpretohet si eksponent në bazën 10. Psh konstantja reale 337.456 mundet gjithashtu të shkruhet si:

3.37456E+2 që ka kuptimin : 3.37456 x 10²

ose ajo mund të shkruhet në forma të tjera si :

0.337456E+3 337.456E0 33745.6E-2 337456.E-3

Një karakter është një nga simbolet e gjuhës C. Me gjithë ndryshimet që mund të ketë nga një variant i gjuhës në tjetrin këto simbole zakonisht përfshijnë shifrat 0 deri në 9; gërmat e mëdha A deri në Z, gërmat e vogla a deri në z, simbolet e pikësimit ; , . ' si dhe simbolet speciale siç janë: +, =, >, <, dhe ↑.

Një konstante karakter konsiston në një karakter të përfshirë ndërmjet dy apostrofeve p.sh:

'A' '+' '3' ':'

N.q.s. një karakter është vetë një apostrof, ai duhet të paraqitet si një çift apostrofesh të përfshira ndërmjet apostrofeve, si : ''

Një varg (zinxhir) karakteresh quhet zakonisht një string. Një konstante string konsiston në një string të përfshirë ndërmjet apostrofeve. Kështu:

'Informatika' dhe 'PDQ123-A'

janë konstante string të vlefshme. Vërejmë përsëri që n.q.s. një apostrof është një nga karakteret e një konstanteje string ai duhet paraqitur si çift apostrofesh p.sh.:

's''mundem'

Per te deklaruar konstantet perdorim sintaksen e meposhtme:

```
# define emri_konstantes vlera_konstantes
```

```
Psh:      #define baza 3
          #define lartesia 7
```

Programi per gjetjen e siperfaqes se drejtkendeshit do te paraqitet me konstante si me poshte:

```
#include <stdio.h> /* deklarimi i konstanteve */
#define baza 3
#define lartesi 7
/* Llogaritja e siperfaqes se nje drejtkendeshi*/
main()
{
    /* deklarimi i variablave */
    int siperfaqja;
    /* pjesa llogaritese */
    siperfaqja = baza * lartesi;
    /* afishimi i rezultatit */
    printf("Baza: %d\n",baza);
    printf("Lartesia: %d\n",lartesi);
    printf("Siperfaqja: %d\n",siperfaqja);
}
```

Variablat

Në matematikë, një simbol shpesh përdoret për t'ju referuar një madhësie. P.sh. formula:

$$S = a.b$$

përdoret për të llogaritur sipërfaqen (e shënuar me S) të një drejtkëndëshi me gjatësi të dhënë (të shënuar me a) dhe gjerësi të dhënë (të shënuar me b). Këta emra simbolikë ose shkurt simbole: S, a dhe b quhen variabla. N.q.s. a-së dhe b-së i jepen vlera të caktuara (konkrete), atëherë ato mund të përdoren nga formula për të llogaritur vlerën e S-së e cila paraqet në fakt sipërfaqen e një drejtkëndëshi të caktuar (konkret).

Kur një variabël përdoret në një program, kompilatori i shoqëron atij një zonë të caktuar memorieje (një ose disa qeliza). Vlera e një variabli në një moment është vlera e regjistruar në këtë zonë memorieje në atë moment, ndërsa emri i tij është i lidhur (simbolizon) adresën e kësaj zone memorieje ku mbahet kjo vlerë.

Emrat e variablave janë identifikatorë prandaj ata u bindet rregullave për të cilat folëm më sipër. Tipi i secilit variabël përcakton edhe tipin e vlerës që mund të regjistrohet në të. Në një program është e detyrueshme të specifikohet tipi i secilit variabël. Kjo bëhet në seksionin e variablave të pjesës së deklarimeve të një program

Supozojme se duam të llogarisim siperfaqen e nje drejtkendeshi. Kemi dy variabla te dhene te plote baza dhe lartesia, prodhimi i te cilave eshte siperfaqja perseri nje variabel i plote.

```
#include <stdio.h>
/* Llogaritja e siperfaqes se nje drejtkendeshi */
```

Gjuha e programimit C

```
main()
{
    /* deklarimi i variablave */
    int baza;
    int lartesia;
    int siperfaqja;
    /* pjesa llogaritese */
    baza = 3;
    lartesia = 7;
    siperfaqja = baza * lartesia;
    /* afishimi i rezultatit */
    printf("%d", siperfaqja);
}
```

Menjehere pas main() deklarohen variablat e nevojshem:

```
int baza; int lartesia; int siperfaqja;
```

Emri i nje variabli tregon adresen ne kujtese, tipi i tij percakton permasat dhe bashkesine e operatoreve qe marrin pjese. Permasat mund te variojne nga lloji i sistemit operativ duke rezervuar per tipin nje hapësire prej dy byte dhe qe lejojne te punosh me numra te plote ne intervalin -32768 deri +32767.

Veprimi: $siperfaqja = baza * lartesia$; I jep variablit siperfaqe prodhimin e vlerave ne variablat baza dhe lartesia.

Veprimi ose instruksioni i fundit `printf("%d",siperfaqja);` afishon 21, vleren e ruajtur ne variablin siperfaqja. Stringu %d midis thonjezave tregon qe vlera numerike e plote e variablit siperfaqja eshte afishuar ne sistemin dhjetor.

Deklarimi i variablave te te njejtit tip mund te behet edhe ne formen:

```
int baza, lartesia, siperfaqja;
```

Ne pergjithesi sintaksa e deklarimit te nje variabli eshte:

Tipi i variablit emri i variablit;

Me qellim qe rezultatet te jene sa me te lexueshem perdorim brenda thonjezave hapësira bosh dhe komente. Psh:

Shembull 1

```
printf("%d ",baza);
printf("%d ",lartesia);
printf("%d",siperfaqe);
```

do marrim si rezultat: 3 7 21

dhe jo 3721

Shembull 2

```
printf("Baza: %d ",baza);
printf("Lartesi: %d ",lartesia);
printf("Siperfaqja: %d",Siperfaqja);
```

do marrim si rezultat:

```
Baza: 3 Lartesi: 7 Siperfaqja: 21
```

Bazat e Informatikës, Programimi në gjuhën C

Tipet e të dhënave

Elementët bazë të një programi janë variablat dhe konstantet. Në C, variablat duhet të deklarohet para se të përdoret dhe në fillim të programit. Ju mund të përcaktoni shumë lloje të variablave numerike. Këtu është një tabelë e llojeve të të dhënave të përcaktuara:

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

Për të deklaruar një variabël mjafton të shkruajmë **tipin** e të dhënës hapësirë bosh **emri i variablit** dhe pikepresje;

Si psh:

int numer; /*ne rast se duam ti japim vlere nje numer te plote */

float numer; /* ne rast se duam ti japim vlere nje numer real ose me presje */

Int, float, double janë numerikë dhe përdoren për të manipuluar numrat dhe tipi char përdoret për të dhënat shkronjore. Tipet e të dhënave të paracaktuara:

Tipi	funksioni	byte
char	Karakter	1
int	nr.l plote	2
float	nr.Real	4
double	Real	8

si dhe tipe të tjera të dhënash:

- Te numurueshem
- Te strukturuar ose rekord
- Union
- Struct

Te cilat do t' trajtojmë në leksionet e tjera në vazhdim.

Instrukcioni i vlerëdhënies

Vlerëdhënia kryhet nepermjet operatorit = .

Ne gjuhën C është e mundur të jepet vlerë disa variablave njëkohësisht. Psh

baza = lartesi = 5;

Ne këtë rast në fillim i jepet vlera 5 variablit lartesi dhe më pas rezultati i jepet variablit baza.

Në këtë paragraf do të përshkruajmë veçoritë e instruksionit përkatës në C.

Instrukcioni i vlerëdhënies ka formën :

variabël = shprehje;

ku shprehje mund të jetë një konstante një variabël tjetër ose një formulë për t'u llogaritur.

Kuptimi i këtij instruksioni është ky : vlera e shprehjes që qëndron në anë të djathtë regjistrohet në zonën e memories me adresë që simbolizohet nga emri i variablit që ndodhet në anën e majtë.

Veprimi **numer = 1000;** i jep vlerën 1000 variablit **numer**.

Veprimet e mëposhtme janë të njëvlerëshme :

(1) **numer = 1000;**

- (2) `numer = 500+500;`
- (3) `numer = (250+250)+500;`

Instrukcioni i hyrjes dhe daljes

Per te afishuar nje vlere ne ekran me funksionin **printf()**, perdorim **%d**. Per te afishuar nje mesazh perdorim **" "**.

Psh: **printf ("vlera e Numerit eshte %d.", numer);**

Funksioni **scanf()** i bibliotekës standarte **stdio.h** përdoret për të lexuar të dhënat nga tastiera.

Paraqitja analitike e printf dhe scanf eshte si me poshte:

Printf(liste_1,liste_2,....,liste_n);

Scanf(liste_1,liste_2,....,liste_n);

Me poshte eshte dhene një shembull qe gjen shumen dy numrave te plote

```
#include<stdio.h>
main()
{
int num1; //deklarojme variablat
int num2;
int shuma;
num1=5; //dhenia vlere fillestare
num2=0;
shuma=0;
printf("Jep nje vlere numrit:\n"); //prompt
scanf("%d", &num2);
shuma=num1+num2; //vleredhenie e shumes
printf("Shuma e numrit te dhene nga tastiera dhe nga perdoruesi eshte: %d",shuma);
//afishon shumen
}
```

Simbolet e karaktereve brenda printf:

- `\n` tregon rresht te ri
- `\t` leviz kursorin nje tab horizontalisht;
- `\r` kthim ne fillim te rreshtit
- `\a` alarm
- `\\` paraqet ne ekran nje `\` ne ekran me `printf`
- `\'` paraqitet `'` ne ekran me `printf`
- `\"` paraqitet `"` ne ekran me `printf`

Kjo tabelë tregon stringje specifike konvertimi te funksioneve `printf()` dhe `scanf()`:

long double	%Lf	%Lf
double	%f	%lf
float	%f	%f
unsigned long int	%lu	%lu
long int	%ld	%ld
unsigned int	%u	%u
int	%d	%d
short	%hd	%hd
char	%c	%c

Operatoret ne gjuhen C

1. Operatoret aritmetike

Operatori	Simboli	Prioriteti
Kllapat	()	Ekzekutohen te parat
Shumezimi	*	Ekzekutohen te dytat
Pjestimi	/	Ekzekutohen te dytat
Mod	%	Ekzekutohen te dytat
Mbledhja	+	Ekzekutohen te fundit
Zbritja	-	Ekzekutohen te fundit

2. Operatoret relacionale

Operatori	Simboli
Barazimi	==
Indryshem	!=
Me I madh	>
Me I vogel	<
Me I madh baraz	>=
Me I vogel baraz	<=

Operatori “=” eshte operator vleredhenie dhe vleresohet nga e djathta ne te majte, kurse operatoret e mesiperme vleresohen nga majta ne te djathte.

3. Operatoret e vleredhenies

	Mbledhja	+=
	Zbritja	-=
	Shumezimi	*=
	Pjestimi	/=
	Mbetja	%=
++a	rrit vleren e a-se me 1 dhe me pas e perdor vleren e a-se	
a++	perdor njehere vleren e a-se, pastaj e rrit me 1	
--a	zbret vleren e a-se me 1 dhe me pas e perdor vleren e a-se	
a--	perdor njehere vleren e a-se, pastaj e zbret me 1	

4. Operatoret llogjike

Operatoret llogjike te algjebres se bool-it perdoren ne shprehjet llogjike komplekse.

Keto jane:

&& (AND llogjik)

|| (OR logico)

! (NOT llogjik)

(kushti_llogjik 1) && (kushti_llogjik 2)

eshte i vertete kur kushti_llogjik 1 dhe kushti_llogjik 2 jane te dy te vertete

(kushti_llogjik 1) || (kushti_llogjik 2)

eshte i vertete kur njeri nga kushtet (kushti_llogjik 1,kushti_llogjik 2) eshte i vertete

```
#include <stdio.h>
void main()
{
int c,c_fill,c_fun;
```

Gjuha e programimit C

```
    c=4; c++;
    printf("c=%d\n",c);
    c=4; ++c;
    printf("c=%d\n",c);
    c_fill=0;
    c_fun = c_fill++;
    printf("c_fun=%d c_fill=%d \n",c_fun, c_fill);
    c_fill =0; c_fun = ++c_fill;
    printf("c_fun=%d c_fill=%d \n",c_fun, c_fill);
}
```

Marrim si rezultat:

```
c=5
c=5
c_fun=0 c_fill=1 /*ne fillim i jepet vlere c_fun=0 dhe me pas c_fill shtohet me 1*/
c_fun=1 c_fill=1 /*shtohet me 1 c_fill dhe me pas I jepet kjo vlere c_fun*/
```

```
/* Ushtrimi.c*/      /* Tipet e te dhenave kryesore int, float, double, char */
/*****/
#include <stdio.h>
void main()
{
int a;
float c;
double d;
char e;
char f[10];
printf("Jepni nje numer te plote\n");
scanf("%d",&a);
printf("Keni futur nje nr. Te plote %d\n\n",a);
printf("jepni nje numer real te tipit float\n");
scanf("%f",&c);
printf("Keni futur nje numer float %f\n\n",c);
printf("jepni nje numer real te tipit double\n");
scanf("%lf",&d);
printf("Keni futur nje numer double %lf\n\n",d);
/* Variablat e tipit karakter marrin vlere ose neperpmjet nje konstante karakter psh: e='A';
ose neperpmjet nje konstante numerike te kodit ASCII psh e=65;*/
printf("Jepni nje karakter\n");
scanf("\n%c",&e);
printf("Keni futur nje karakter %d\n\n",e);
/* kodi i formatimit per nje string %s */
printf("jepni nje string (max 10 karaktere)\n");
scanf("%s",&f);
printf("keni futur nje string %s\n\n",f);
}
```