

Tema 2

Operatori, naredbe selekcije i
repeticije

Operatori

- Uzimaju vrednosti u vidu operandata i na osnovu njih sračunavaju rezultat operacije
 - Mogu biti unarni, binarni i ternarni na osnovu broja operandata
- Grupe operatora
 - Aritmetički
 - Relacioni
 - Logički
 - Dodele vrednosti
 - Adresni
 - Bit-operatori

Redosled izvršavanja operatora

- Definisan u jeziku pomoću nivoa prioriteta
- Kao i u matematici, zagrade (i) imaju najviši prioritet
- U slučaju postojanja više izraza sa istim prioritetom, izvršavanje ide u smeru grupisanja operatora na tom nivou prioriteta
 - s leva na desno ili s desna na levo

Programski jezici i strukture podataka - Tema 2

Tabela prioriteta i smera grupisanja operacija

Prioritet	Broj operandata	Operatori	Smer
15	2	[] () . ->	➡
14	1	! ~ ++ -- + - * & (tip) sizeof	⬅
13	2	* %	➡
12	2	+ -	➡
11	2	<< >>	➡
10	2	< <= > >=	➡
9	2	== !=	➡
8	2	&	➡

Programski jezici i strukture podataka - Tema 2

Tabela prioriteta i smera grupisanja operacija

Prioritet	Broj operanada	Operatori	Smer
7	2	\wedge	\rightarrow
6	2	$ $	\rightarrow
5	2	$\&\&$	\rightarrow
4	2	$ $	\rightarrow
3	3	$? :$	\rightarrow
2	2	$= \quad += \quad -= \quad *= \quad /= \quad \% =$ $\&= \quad ^= \quad = \quad <<= \quad >>=$	\leftarrow
1	2	$,$	\rightarrow

Primer 1

- Aritmetičke operacije

```
#include <stdio.h>

int main() {
    int a = 5;
    int b = 3;
    printf("Zbir a + b je : %d\n", a + b);
    printf("Razlika a - b je : %d\n", a - b);
    printf("Proizvod a * b je : %d\n", a * b);
    printf("Celobrojni kolicnik a / b je : %d\n", a / b);
    printf("Lose racunanje realnog kolicnika a / b je : %f\n",
           a / b);
    printf("Realni kolicnik a / b je : %f\n",
           (float)a / (float)b);
    printf("Ostatak pri deljenju a / b je : %d\n", a % b);
    return 0;
}
```

Aritmetički operatori i operatori dodele vrednosti

- Zagrade ()
- Dodela vrednosti =
- Sabiranje +
- Oduzimanje -
- Množenje *
- Deljenje /
- Deljenje po modulu %
- Inkrementacija (uvećanje operanda za 1) ++
- Dekrementacija (smanjenje operanda za 1) --

Primeri korišćenja aritmetičkih i operatora dodele vrednosti

- $a = a + 3 \Leftrightarrow a += 3$
- $a = a - b \Leftrightarrow a -= b$
- $a = a * c \Leftrightarrow a *= c$
- $/=, \%=...$

- $a = a + 1 \Leftrightarrow a += 1 \Leftrightarrow a++ \Leftrightarrow ++a$
- $b = b - 1 \Leftrightarrow b -= 1 \Leftrightarrow b-- \Leftrightarrow --b$

- $i = j++; \Leftrightarrow i = j; j = j + 1;$
- $i = ++j; \Leftrightarrow j = j + 1; i = j;$

Relacioni i logički operatori

- Sve vrednosti različite od 0 su logički tačne vrednosti
- Vrednost 0 je logički netačna vrednost
- Relacioni operatori: <, >, <=, >=, ==, !=
- Logički operatori:
 - && logičko "I"
 - || logičko "ILI"
 - ! logičko "NE" (negacija)

Pitanje

Koja će biti vrednost promenljive `x` nakon sledećih naredbi:

- `x = (5 == 5) == 5 ;`
- `x = -7 && 5 ;`

Podrazumevati da je rezultat 1 za logički tačnu vrednost.

Zadatak 1

Napisati program koji od korisnika traži da unese tri celobrojne vrednosti, na osnovu kojih će se izračunati, a zatim ispisati njihova aritmetička, harmonijska, geometrijska i kvadratna sredina.

- Formule za tražene sredine slobodno potražiti na Internetu

Iskazi selekcije

- Omogućavaju da se grupa naredbi izvrši uslovno, u zavisnosti od vrednosti iskaza
- Naredbe grananja
 - `if (else-if)`
 - `switch`

if-else iskaz

```
if(izraz)
    naredba_1
else
    naredba_2
```

- Ako je izraz logički tačan, izvršava se naredba1, inače naredba2
- Može se navesti blok naredbi (uokviren { i } zagradama)
- moguće je ulančati više if-else-if iskaza

```
if(izraz_1)
    naredba_1
else if(izraz_2)
    naredba_2
else if(izraz_3)
    ...
else
    naredba_n
```

Ternarni operator

- `izraz ? naredba1 : naredba2`
- funkcioniše na identičan način kao `if-else`

```
min = (x < y) ? x : y ;
```

Zadatak 1

- a Napisati C program koji traži od korisnika da unese dva broja. Ispisati . veći od ta dva broja.
- b Napisati C program koji traži od korisnika da unese tri broja. Ispisati . najmanji od ta tri broja.

Zadatak 2

Napisati program kojim se za zadato x izračunava $f(x)$ po formuli:

$$f(x) = \begin{cases} -7, & x < 0 \\ x^{1/4} + 4, & 0 \leq x < 1 \\ 2\sqrt{x} - 5, & 1 \leq x < 13 \\ x^2/9, & x \geq 13 \end{cases}$$

Zadatak 3

Napisati program koji učitava vrednosti promenljivih a i b , a zatim rešava jednačinu:

$$ax + b = 0$$

- Pokriti sve slučajeve za različite kombinacije vrednosti a i b

Zadatak 4

Tri tačke su zadate koordinatama u ravni. Napisati program kojim se izračunava površina paralelograma čiji su vrhovi date tačke.

- Obratiti pažnju na proveru da li tačke u ravni formiraju trougao, odnosno paralelogram

Zadatak 5

Napisati program koji proverava da li se od unetih vrednosti uglova može formirati trougao. Ukoliko je to moguće, analizirati unete uglove:

- Ako su dva ugla jednaka, ispisati da se radi o jednokrakom trouglu
- Ako je jedan ugao od 90 stepeni, ispisati da se radi o pravouglom trouglu
- Ako su sva tri ugla jednaka (60 stepeni), ispisati da se radi o jednakokrakom trouglu
- Za sve ostale slučajeve, samo ispisati da se može formirati trougao

Napomena:

Ukoliko neki trougao ispunjava više od jedne osobine, ispisati sve osobine koje zadovoljava.

Zadatak 6

Napisati program koji za unetu visinu (u centimetrima) i telesnu težinu osobe (u kilogramima) ispisuje njen BMI indeks.

$$BMI = \frac{masa[kg]}{visina^2[m]}$$

Potom ispisati određeni ispis na ekran na osnovu dobijene vrednosti indeksa:

- Veoma neuhranjeno, za vrednosti manje od 16.5
- Neuhranjeno, za vrednosti između 16.5 i 18.5
- Normalno, za vrednosti od 18.5, do 24.9
- Prekomerno, za vrednosti od 25, do 29.9
- Gojaznost, za vrednosti veće od 30

switch iskaz

- Višestruki uslovni iskaz
 - Sličan if-else-if konstrukciji
- Izbor proizvoljnog broja opcija
 - opcija je isključivo celobrojna vrednost

```
switch(izraz) {  
    case const-izraz:  
        naredbe  
    case const-izraz:  
        naredbe  
    ...  
    default:  
        naredbe  
}
```

Primer 2

Upotreba switch iskaza

```
#include <stdio.h>

int main() {
    int ocena;

    printf("Unesite ocenu: ");
    scanf("%d", &ocena);

    switch(ocena) {
        case 5:
            printf("Odlican\n");
            break;
        case 4:
            printf("Vrlo dobar\n");
            break;
        case 3:
            printf("Dobar\n");
            break;
```

Programski jezici i strukture podataka - Tema 2

```
    case 2:
        printf("Dovoljan\n");
        break;
    case 1:
        printf("Nedovoljan\n");
        break;
    default:
        printf("Unesite ocenu izmedju 1 i 5!\n");
}

return 0;
}
```

Naredba break

- Blok naredbi se napušta bez obzira na trenutnu vrednost uslova
- Program nastavlja sa izvršavanjem prve naredbe posle bloka
- U slučaju više blokova koji su jedan u drugom (ugneždeni), izlazi se samo iz onog gde se `break` naredba nalazi

while ciklus

```
while(izraz) {  
    naredbe  
}
```

- Naredbe se izvršavaju sve dok je izraz logički tačan
- Kada to više nije slučaj, izlazi se iz petlje

Primer 1

```
#include <stdio.h>

int main() {
    float fahr, celsius;
    float lower, upper, step;

    lower = 0;           /* lower limit of temperature scale */
    upper = 300;          /* upper limit */
    step = 20;            /* step size */

    fahr = lower;
    while (fahr <= upper) {
        celsius = (5.0 / 9.0) * (fahr - 32.0);
        printf("%3.0f %6.1f\n", fahr, celsius);
        fahr += step;
    }

    return 0;
}
```

for ciklus

- Brojačka promenljiva kontroliše izvršavanje petlje
 1. Inicijalizacija (samo prvi put)
 2. Logička provera izraza (kontroliše dalje izvršavanje petlje)
 3. Promena brojačke promenljive (nakon svake iteracije petlje)
- Prilikom svake iteracije, izvršava se blok naredbi

```
for (i = 0; i <= 50; i++) {  
    printf("%d\n", i);  
}
```

- Različiti slučajevi mogu se lepše zapisati pomoću `while`, odnosno, `for` ciklusa
 - `for` kada je unapred poznat broj iteracija
 - `while` kada nije poznat broj, već izraz čiji logički rezultat određuje sledeću iteraciju

Zadatak 1

Napisati C program koji računa sumu prvih n prirodnih brojeva, pri čemu se n zadaje na početku programa.

Zadatak 2

Realizovati množenje i deljenje pomoću sabiranja i oduzimanja.

do-while ciklus

```
do {  
    naredbe  
} while(izraz);
```

- Izraz se izračunava nakon tela ciklusa
- Telo ciklusa će se sigurno barem jednom izvršiti
 - Makar izraz odmah bio netačan, jer se proverava posle tela ciklusa

Upotreba do-while ciklusa

- Kontrolisanje unosa korisnika
- Postaviti na mesto izraza svojstvo koje ne sme biti zadovoljeno
 - Negacija od očekivanog korisničkog unosa
 - Demorganov zakon?

```
do {  
    printf("Unesite cifru: ");  
    scanf("%d", &n);  
} while(<izraz>);
```

Formulisati izraz u do-while petlji tako da korisnik može uneti:

- Prirodan, paran broj
- Negativan broj, deljiv sa tri

Upotreba break i continue u ciklusima

- Naredba `break` omogućava da se prekine izvršavanje petlje
- Naredba `continue` omogućava da se odmah pređe na narednu iteraciju petlje

```
for(i = 1; i <= 10; i++) {  
    if(i == 5) {  
        <naredba>  
    }  
    printf("%d\n", i);  
}
```

Koje će se cifre ispisati na ekran ako umesto naredbe u kodu:

- Stoji `break` naredba
- Stoji `continue` naredba

Zadatak 3

Napisati program kojim se za celobrojnu vrednost n izračunava njegov faktorijel.

$$n! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot n$$

Zadatak 4

Sa standardnog ulaza učitati prirodne brojeve N i q . Koristeći `while` ciklus ispisati sve brojeve od 2 do N koji su deljivi sa q .

Zadatak 5

Sa standardnog ulaza učitati prirodan broj N . Ispisati sve njegove činioce.

Zadatak 6

Napisati program koji ispituje da li je dati prirodan broj n prost. Broj je prost ako je deljiv samo sa jedan i sa samim sobom.

Zadatak 7

Napisati program kojim se štampaju svi trocifreni Armstrongovi brojevi. Broj je Armstrongov ako je jednak zbiru kubova svojih cifara.

Zadatak 8

Napisati program koji obrće cifre bilo koje celobrojne vrednosti.

Na primer, za broj 1234, rezultat nakon obrtanja je 4321.

Zadatak 9

Po čuvenoj priči, profesor je zadao učenicima zadatak da saberu prvih 100 prirodnih brojeva. Mladi Gaus je primetio da postoji obrazac i vrlo brzo rešio zadatak, tako što je sabirao 1 i 99, 2 i 98 itd.

Napisati program koji računa zbir prvih sto brojeva koristeći Gausovu metodu. Ispisati parove brojeva tokom računanja na sledeći način:

```
1 + 99 = 100
2 + 98 = 100
...
+ 50
+ 100
----
5050
```