

Tema 2

char tip, selekcija, repeticija i nizovi

char tip

- Karakteri: 'A', 'B', 'c', 'd', '1', '4', '#', '?', '\n'
- char varijabla se koristi za čuvanje tekstualnih znakova:
 - slova,
 - cifara,
 - specijalnih znakova,
 - nešampajućih (belih) znakova,
 - ali može se koristiti i za čuvanje malih celobrojnih vrednosti (0 do 255 ili -128 do 127).

char je samo broj!

- Svakom karakteru se pridružuje numerički kod.
- Postoje različiti skupovi kodova:
 - ASCII (American Standard Code for Information Interchange) – najčešći.
 - EBCDIC – zastareo, danas se retko koristi.
 - Noviji skupovi karaktera (Unicode).
- Koristićemo ASCII.

Primer 1

/* koristenje char kao znakovnog tipa i za malu numericku vrednost */

```
#include<stdio.h>
int main()
{
    char znak;
    printf("Unesite znak: ");
    scanf("%c", &znak);
    printf("Znak kao karakter je: %c\n", znak);
    printf("Numericka vrednost znak-a je: %d\n", znak);
    printf("Karakter posle %c je %c\n", znak, znak+1);
    return 0;
}
```

Iskazi selekcije

- Omogućava da se određene naredbe izvrše uslovno u zavisnosti od vrednosti određenog iskaza
- Iskazi selekcije (grananje):
 - if
 - switch

if-else iskaz

if (*izraz*)
 *naredba*₁
else
 *naredba*₂

- Ako je *izraz* tačan, *naredba*₁ se izvršava.
- Ako je je *izraz* netačan, *naredba*₂ se izvršava
- U oba slučaja umesto naredbi može se navesti blok naredbi

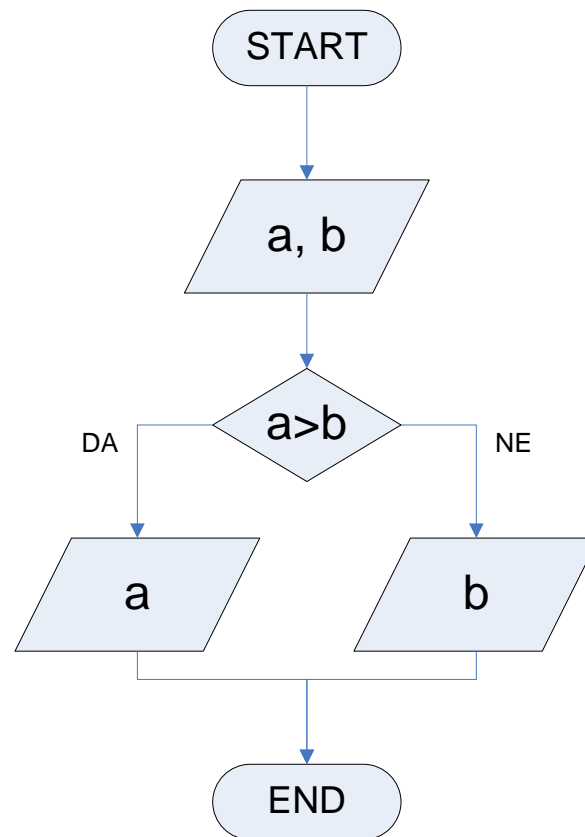
Ternarni operator

- condition ? expression1 : expression2
- Funkcioniše kao if-else

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

Zadatak 1

- Napisati C program koji ispisuje veći od dva cela broja.



switch iskaz

- Višestruki uslovni iskaz
 - Sličan if-else if-else konstrukciji
- Omogućava izbor proizvoljnog broja opcija. Izbor je zasnovan na celobrojnoj vrednosti.

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

Primer 2

```
switch (ocena)
{
    case 5:      printf("Odlican!\n\n");
                 break;
    case 4:      printf("Vrlo dobar!\n\n");
                 break;
    case 3:      printf("Dobar!\n\n");
                 break;
    case 2:      printf("Dovoljan!\n\n");
                 break;
    case 1:      printf("Nedovoljan!\n\n");
                 break;
    default:     printf("Ocena mora biti izmedju 1 i 5.");
}
}
```

break

- Kada se naiđe na **break**, blok naredbi se napušta bez obzira na trenutnu vrednost uslova.
- Program nastavlja sa izvršavanjem prve naredbe iza bloka.
- Ukoliko se pozove u ugnježdenim blokovima, **break** izlazi samo iz unutrašnjeg bloka.

Zadatak 2

- Realizovati prethodni zadatak uz pomoć if selekcije.

```
if (izraz) {  
    naredba1;  
} else if (izraz2) {  
    naredba2;  
} else if (izraz3) {  
    naredba3;  
} else {  
    naredba4;  
}
```

Zadatak 3

- 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 4

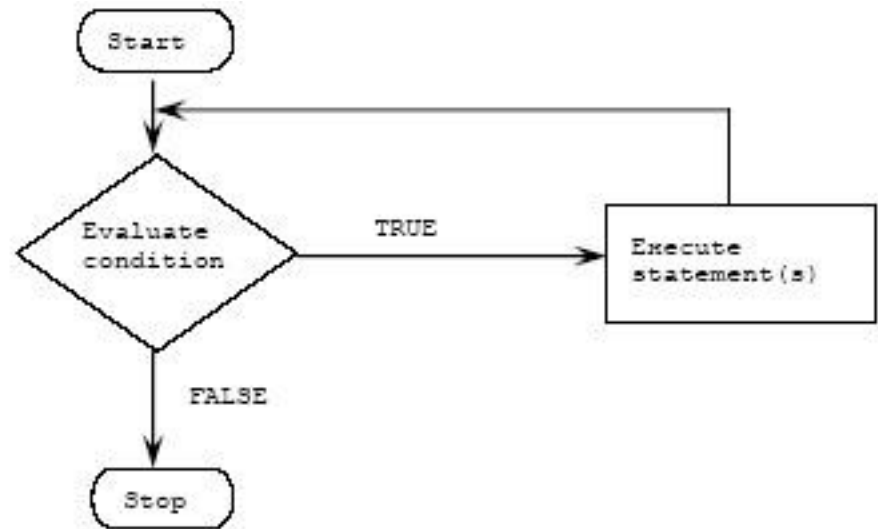
- Napisati program koji učitava vrednosti promenljivih **a** i **b**, a zatim rešava jednačinu **$ax+b=0$** .
 - pokriti sve slučajeve za različite kombinacije vrednosti **a** i **b**

Zadatak 5

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

while ciklus

```
while (uslov) {  
    naredbe;  
}
```



- Naredbe se izvršavaju sve dok je *uslov* ispunjen, kada to više nije slučaj izlazi se iz petlje

Primer 3

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

```
/* print Fahrenheit-Celsius table for fahr = 0, 20, ..., 300; floating-point version */
```

```
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 = fahr + step;
```

```
}
```

```
    return 0;
```

```
}
```

for ciklus - primer

- For petlje se kontrolišu brojačkom promenljivom.
- Primer

**i je kontrolna
promenljiva - brojač.**

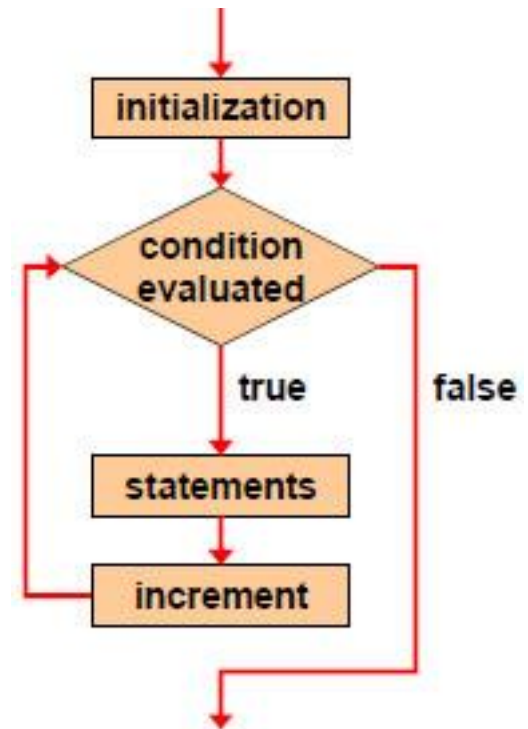
```
for ( i =0; i<=50; i++) {  
    printf(“%d\n”, i);  
}
```

**i se inkrementira nakon
svake iteracije (može se
i dekrementirati – po
potrebi)**

for i while

for(initialization; expression; increment)
 statement

```
initialization;  
while( expression )  
{  
    statement  
    increment;  
}
```



Zadatak 6

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

Zadatak 7

Realizovati množenje i deljenje pomoću sabiranja i oduzimanja.

do – while ciklus

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

- Sličan while ciklusa
 - Uslov se izračunava **nakon** tela ciklusa
 - Telo ciklusa se izvršava NAJMANJE jednom, čak i kada je uslov odmah netačan (jednak nuli)

do – while ciklus

```
int broj;
```

```
do {
```

```
    printf("Unesite pozitivan ceo broj: ");
```

```
    scanf("%d", &broj);
```

```
} while ( broj <= 0 );
```

do – while ciklus

```
int broj;
```

```
do {
```

```
    printf("Unesite negativan paran broj : ");
```

```
    scanf("%d", &broj);
```

```
} while ( ##### );
```

```
// ##### = ?
```


do – while ciklus

```
int broj;
```

```
do {
```

```
    printf("Unesite neparan broj deljiv sa 3 : ");
```

```
    scanf("%d", &broj);
```

```
} while ( ##### );
```

```
// ##### = ?
```

break i continue

- *break* omogućava da se prekine izvršavanje petlje
- *continue* omogućava da se pređe na narednu iteraciju

Zadatak 8

Napisati program kojim se za dato n izračunava faktorijel prirodnog broja n . Koristeći do-while napraviti zaštitu za unos broja n :

$$\text{fakt} = n! = 1 * 2 * 3 * \dots * n$$

Zadatak 9

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

Zadatak 10

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 11

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

Zadatak 12

Za svaku od navedenih formula napisati program koji računa broj π (preciznost zavisi od dužine niza):

$$\text{a)} \quad \frac{\pi^2}{6} = \frac{1}{1^2} + \frac{1}{2^2} + \frac{1}{3^2} + \frac{1}{4^2} + \dots$$

$$\text{b)} \quad \frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \dots$$

$$\text{c)} \quad \frac{\pi}{2} = \frac{2}{1} * \frac{2}{3} * \frac{4}{3} * \frac{4}{5} * \frac{6}{5} * \frac{6}{7} * \dots$$

$$\text{d)} \quad \pi = 3 + \frac{4}{2*3*4} - \frac{4}{4*5*6} + \frac{4}{6*7*8} - \frac{4}{8*9*10} + \dots$$

Zadatak 13*

Napisati program koji za dati prirodan broj n izračunava:

$$S = 1 * 2 + 2 * 3 * 4 + \dots + (n+0) * (n+1) * \dots * (n+n).$$

Nizovi - deklaracija

- Niz predstavlja kolekciju elemenata istog tipa.
Primer deklaracije niza je:
- **int niz[5];**
/* niz od 5 elemenata tipa int*/

Pristupanje elementima niza

- Pristupanje elementima niza se ostvaruje na sledeći način:
- niz[0] = 4;
- niz[1] = 2 * niz[0]; /* niz[1] = 8 */
- niz[2] = niz[0] * niz[1]; /* niz[2] = 32 */
- niz[3] = 5;
- niz[4] = 7;
- a = niz[10];
- Napomena: Indeks niza ide od 0 (a ne od 1!) do n-1

Inicijalizacija niza

- Primer inicijalizacije vrednosti elemenata niza

```
for ( i=0; i<5; i++)  
    niz[i] = 0;
```

Inicijalizacija niza

- Opšta sintaksa:

`tip_podatka ime[veličina] = {vred0, vred1, ... vredn-1};`

- Primer:

`int dani_u_mesecu[12] = {31, 28, 31, 30, 31, 31, 30, 31, 30, 31};`

- Broj elemenata za inicijalizaciju može da bude manji od veličine niza.
 - ostatak se inicijalizuje nulama

- Primer:

`int dani_u_mesecu[12] = {31, 28, 31, 30, 31, 31, 30};`

Inicijalizacija niza

- Dimenzija niza se može izostaviti ako se niz inicijalizuje
 - veličina niza je broj elemenata za inicijalizaciju
- Primer:
`int dani_u_mesecu[] = {31, 28, 31, 30, 31, 31, 30, 31, 30, 31};`

Primer 4

Dat je niz od maksimalno 30 celobrojnih elemenata. Učitati n elemenata i ispisati ih po učitanom i obrnutom redosledu.

```
1  #include <stdio.h>
2  #define MAX_SIZE 30
3
4  int main()
5  {
6      int a[MAX_SIZE];
7      int i, n;
8
9      do {
10         printf("Unesite broj elemenata niza (maksimalno %d): ", MAX_SIZE);
11         scanf("%d", &n);
12     } while ( n <= 0 || n>MAX_SIZE);
13
14     for (i = 0; i<MAX_SIZE; i++) {
15         a[i] = 0;
16     }
17
18     for (i = 0; i<n; i++) {
19         printf("a[%d]=", i);
20         scanf("%d", &a[i]);
21     }
22
23     printf("\n Elementi niza po ucitanom redosledu: \n");
24
25     for (i = 0; i<n; i++) {
26         printf("a[%d]=%d\n", i, a[i]);
27     }
28
29     printf("\n Obrnuti redosled: \n");
30
31     for (i = n-1; i>=0; i--) {
32         printf("a[%d]=%d\n", i, a[i]);
33     }
34
35     return 0;
36 }
```

Zadatak 14

Dat je niz A od maksimalno 30 celobrojnih elemenata.
Ucitati n elemenata, zatim ucitati ceo broj br.
Na izlazu ispisati ceo niz i broj pojavljivanja broja br u nizu.

primer.

$A=[2, 5, 6, 2, 8, 9, 2]$

$br=2$

resenje:

"Broj 2 se pojavljuje 3 puta u nizu $A=[2, 5, 6, 2, 8, 9, 2]$ "

Zadatak 15

Dat je niz od maksimalno 20 realnih elemenata. Učitati n elemenata, a zatim naći maksimalnu vrednost.

Zadatak 16

Dat je niz X od maksimalno 25 celobrojnih elemenata. Učitati n elemenata u niz X i formirati nizove A i B, pri čemu su elementi niza A parni elementi niza X, a elementi niza B su negativni elementi niza X. Ispisati nizove X, A i B.

Zadatak 17

Dat je niz C od maksimalno 80 karaktera. Učitati n karaktera u niz C i formirati niz A, pri čemu su elementi niza A samo slova iz niza C.

Zadatak 18

Napisati program koji pronalazi prvi element niza koji je najbliži srednjoj vrednosti niza celih brojeva. Niz može da ima najviše 20 elemenata.