

# PROGRAMIRANJE 1

# PROGRAMSKI JEZIK

# C

Staša Vujičić  
Čas 8

# NAREDBA SWITCH

- Naredba switch se naziva još i naredba višestrukog grananja.
- Ovom naredbom se proverava da li je neki izraz jednak jednoj od više konstantnih celobrojnih vrednosti i u zavisnosti od toga, preduzima se odgovarajuća akcija.

# NAREDBA SWITCH

- Opšti oblik ove naredbe je:

```
switch (izraz) {  
  case konstantan_izraz1: naredbe1  
  case konstantan_izraz2: naredbe2  
  ...  
  default: naredbe  
}
```

- Svaki konstantni izraz predstavlja određeni slučaj. Ako neki slučaj odgovara vrednosti izraza, izvršavanje počinje od tog slučaja.
- Svi izrazi slučajeva moraju biti različiti.
- Slučaj označen sa default nije obavezan. Ako se ne navede i nijedan drugi slučaj ne odgovara vrednosti izraza onda se neće izvršiti nikakva akcija.

# NAREDBA SWITCH

```
#include<stdio.h>
int main()
{
    int n;

    printf("Unesi paran broj manji od 10\n");
    scanf("%d",&n);

    switch(n)
    {
    case 0:
        printf("Uneli ste nulu\n");
    break;
```

case 2:

```
printf("Uneli ste dvojku\n");
```

break;

case 4:

```
printf("Uneli ste cetvorku\n");
```

break;

case 6:

```
printf("Uneli ste sesticu\n");
```

break;

case 8:

```
printf("Uneli ste osmicu\n");
```

break;

default:

```
printf("Uneli ste nesto sto nije paran  
broj\n");
```

```
} }
```

# NAREDBA SWITCH

- ◉ *Napisati program koji vrši brojanje pojavljivanja karaktera 0, 1 i 2 korišćenjem switch naredbe.*

- ◉ `#include <stdio.h>`

```
main()
```

```
{
```

```
    int c;
```

```
    int br_0=0, br_1=0, br_2=0;
```

```
while ((c = getchar()) != EOF)
{
    switch(c)
    {
        /* Obratiti paznju da nije case 0: niti case '0'; */
        case '0':
            br_0++;
            break; /* Isprobati veziju bez break */
        case '1':
            br_1++;
            break;
        case '2':
            br_2++;
            break;
    }
}
printf("Br 0 : %d\nBr 1 : %d\nBr 2 : %d\n",br_0, br_1, br_2);
}
```



# NAREDBA SWITCH

## ◉ Jednostavan kalkulator

Program učitava komandu sa ulaza (u vidu jednog karaktera) i na osnovu te komande učitava dva broja nad kojima izvršava odgovarajuću operaciju.

Ovaj postupak se ponavlja dok se ne unese komanda za kraj programa.

Program ilustruje primenu *do-while petlje* i *switch naredbe* za implementaciju menija, za potrebe interaktivnog rada.

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

```
/* Program ilustruje switch() naredbu */
```

```
main()
```

```
{
```

```
char c;
```

```
int a,b;
```

```
/* Ova petlja predstavlja tipican nacin za  
implementaciju interaktivnog menija:
```

- u svakoj iteraciji se ispisuje meni,
- od nas se ocekuje da unesemo karakter za komandu.

U zavisnosti od unetog karaktera se preduzima odgovarajuca akcija (u ovom slucaju unos dva broja i prikaz rezultata odgovarajuce operacije).

Petlja se izvrsava sve dok se ne unese komanda za kraj ('q' u ovom slucaju). \*/

do

{

```
/* Prikaz menija */  
printf("Unesite:\n\t  
    'a' za sabiranje,\n\t  
    's' za oduzimanje,\n");  
printf("\t'm' za mnozenje,\n\t  
    'd' za deljenje,\n\t  
    'q' za kraj.\n");
```

```
/* Ucitavamo komandu */  
scanf(" %c", &c);
```

```
/* Izbor akcije na osnovu komande koja je uneta */
```

```
switch(c)
```

```
{
```

```
    /* Slucaj sabiranja */
```

```
    case 'a':
```

```
        printf("Unesite dva cela broja: ");
```

```
        scanf("%d%d", &a, &b);
```

```
        printf("%d+%d=%d\n", a, b, a + b);
```

```
        break;
```

```
    /* Slucaj oduzimanja */
```

```
    case 's':
```

```
        printf("Unesite dva cela broja: ");
```

```
        scanf("%d%d", &a, &b);
```

```
        printf("%d-%d=%d\n", a, b, a - b);
```

```
        break;
```

```
/* Slucaj mnozenja */
```

```
case 'm':
```

```
    printf("Unesite dva cela broja: ");
```

```
    scanf("%d%d", &a, &b);
```

```
    printf("%d*%d=%d\n", a, b, a * b);
```

```
    break;
```

```
/* Slucaj deljenja */
```

```
case 'd':
```

```
    printf("Unesite dva cela broja: ");
```

```
    scanf("%d%d", &a, &b);
```

```
    printf("%d/%d=%d\n", a, b, a / b);
```

```
    break;
```

```
/* Slucaj izlaska -- ne radimo nista */
```

```
    case 'q':
```

```
        break;
```

```
/* Slucaj pogresne komande (sve ostalo sto  
nije navedeno gore) */
```

```
    default:
```

```
        printf("Pogresna komanda!\n");
```

```
        break;
```

```
    }
```

```
} while(c != 'q'); /* Iz petlje izlazimo kada je c  
== 'q' */
```

```
}
```

# NAREDBA SWITCH

- Broj reči na ulazu

Program učitava tekst sa ulaza, i pri tom broji reči.

Pod rečju se podrazumeva niz ne-blanko karaktera (blanko karakteri su ' ', '\n' i '\t'). Program ilustruje *switch naredbu*.

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

```
#define IN 1      /* Ako smo trenutno u reci */
```

```
#define OUT 0    /* Ako smo trenutno van reci */
```

```
/* Program ilustruje switch() naredbu */
```

```
main ()
```

```
{
```

```
    int stanje = OUT; /* Stanje: u ili van reci */
```

```
    int br_reci = 0; /* Brojac reci */
```

```
    int c;
```

```
    /* Citamo karakter po karakter */
```

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

```
    {
```



```
/* U zavisnosti od stanja preduzimamo akcije
*/
```

```
switch (stanje)
```

```
{
```

```
/* Ako smo u stanju OUT... */
```

```
case OUT:
```

```
/* Proveravamo koji je karakter ucitan */
```

```
switch (c)
```

```
{
```

```
/* Ako je belina ne radimo nista */
```

```
case ' ':
```

```
case '\t':
```

```
case '\n':
```

```
break;
```

```
/* U ostalim slucajevima smo na pocetku nove  
reci, pa je brojimo, a zatim prelazimo u  
stanje IN, koje oznacava da se trenutno  
nalazimo u reci. */
```

```
default:
```

```
br_reci++;
```

```
stanje = IN;
```

```
break;
```

```
}
```

```
break; /*Kraj slucaja OUT*/
```

```

/* Ako smo u stanju IN... */
  case IN:
    /* Proveravamo koji je karakter ucitan */
    switch (c)
    {
      /* Ako je belina, tada prelazimo u stanje
         OUT, koje oznacava da smo van reci. */
      case ' ': case '\t': case '\n':
        stanje = OUT;
        break;
      /* U ostalim slucajevima ne radimo nista */
      default:
        break;
    }
    break; /*Kraj slucaja IN*/
  }
}

```

```
/* Ispisujemo broj reci */  
printf ("Broj reci je: %d\n", br_reci);  
  
}
```

# USLOVNI IZRAZ

- Uslovni izrazi obezbeđuju alternativni način pisanja if-else konstrukcije.

- Tako se

```
if (izraz1) izraz2;  
    else izraz3;
```

- može zapisati kao

```
izraz1 ? izraz2 : izraz3;
```

- Primer korišćenja uslovnog izraza:

```
min = (a<b)? a : b;
```

```
max = (a>b)? a : b;
```

# USLOVNI IZRAZ

- Brojanje malih i velikih slova

Program čita tekst sa ulaza, prevodi sva mala slova u velika, i pri tom broji mala i velika slova.

Program ilustruje operatore ',' (zarez), i '?:' (uslovni operator).

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

```
/* Program ilustruje operatore ?: i , (zarez) */
```

```
main()
```

```
{
```

```
    int c;
```

```
    int br_malih;
```

```
    int br_velikih;
```

```
/* Prepisujemo ulaz na izlaz, uz promenu malih slova u velika. Takodje, brojimo mala i velika slova. */
```

```
for(br_malih = 0, br_velikih = 0 ; (c = getchar())  
!= EOF ; )
```

```
{
```

```
    putchar(c >= 'a' && c <= 'z' ? c - 'a' + 'A' : c);
```

```
    if(c >= 'a' && c <= 'z') br_malih++;
```

```
    if(c >= 'A' && c <= 'Z') br_velikih++;
```

```
}
```

```
printf("Broj velikih slova: %d\nBroj malih slova:  
%d\n",
```

```
    br_velikih, br_malih);
```

```
}
```