

Programiranje 1

Beleške sa vežbi

Smer *Informatika*

Matematički fakultet, Beograd

Jelena Tomašević

December 4, 2005

Sadržaj

1		2
1.1	Konverzije	2
1.2	Makroi	4

Čas 1

1

1.1 Konverzije

Primer 1 `atoi` - konverzija niske cifara u brojnu vrednost. Program omogućuje čitanje celog broja bez korišćenja funkcije `scanf("%d")`.

Napomena : funkcija `atoi` već postoji u zaglavlju `<stdio.h>` i ovde je implementirana samo za vežbu.

```
#include <stdio.h>
#include <ctype.h> /* Zbog funkcije isspace */

/* Maksimalna duzina reci */
#define MAX_DIGITS 10

/* AsciiTInteger
Funkcija izracunava vrednost celog broja koji je zapisan u datom
nizu karaktera. Za izracunavanje se koristi Hornerova shema. */

int atoi(char s[]) {
    int sum = 0;

    /* Obradjuju se karakteri sve dok su cifre */
    int i;
    for (i = 0; isdigit(s[i]); i++)
        sum = 10*sum + (s[i]-'0');

    return sum;
}

main() {
    char s[MAX_DIGITS];
    int num;
```

¹Zasnovano na primerima sa sajtova <http://www.matf.bg.ac.yu/~filip>,
<http://www.matf.bg.ac.yu/~milena>, <http://www.matf.bg.ac.yu/~jelenagr>.

```
/* Ucitavamo broj u vidu niske karaktera */
printf("Unesi ceo broj : ");

/* Obratiti paznju da uz s ne stoji & */
scanf("%s", s);

/* Izracunava se njegova vrednost */
num = atoi(s);

/* Ispisuje se vrednost */
printf("Uneo si broj : %d\n", num);
printf("Broj za tri veci je : %d\n", num+3);
}
```

```
Ulaz:
Unesi ceo broj : 5
Izlaz:
Uneo si broj : 5
Broj veci za tri je : 8
```

Primer 2 *btoi - konverzija iz datog brojnog sistema u dekadni.*

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

/* Pomocna funkcija koja izracunava vrednost koju predstavlja
karakter u datoj osnovi
Funkcija vraca -1 ukoliko cifra nije validna.

Npr.
cifra 'B' u osnovi 16 ima vrednost 11
cifra '8' nije validna u osnovi 6

*/

int digit_value(char c, int base) {
    /* Proveravamo obicne cifre */
    if (isdigit(c) && c < '0'+base)
        return c-'0';

    /* Proveravamo slovne cifre za mala slova */
    if ('a'<=c && c < 'a'+base-10)
        return c-'a'+10;

    /* Proveravamo slovne cifre za velika slova */
    if ('A'<=c && c < 'A'+base-10)
        return c-'A'+10;

    return -1;
}
```

```

}

/* Funkcija izracunava vrednost celog broja koji je zapisan u
datom
nizu karaktera u datoj osnovi. Za izracunavanje se koristi Hornerova shema.
*/ int btoi(char s[], int base) {
    int sum = 0;

    /* Obradjuju se karakteri sve dok su cifre */
    int i, vr;
    for (i = 0; (vr = digit_value(s[i], base)) != -1; i++)
        sum = base*sum + vr;

    return sum;
}

main() {
    char bin[] = "11110000";
    char hex[] = "FF";

    printf("Dekadna vrednost binarnog broja %s je %d\n", bin, btoi(bin, 2));
    printf("Dekadna vrednost heksadekadnog broja %s je %d\n", hex, btoi(hex, 16));
}

```

Izlaz:

Dekadna vrednost binarnog broja 11110000 je 240

Dekadna vrednost heksadekadnog broja FF je 255

1.2 Makroi

Primer 3 *Šta je rezultat rada sledeceg programa :*

```

#include <stdio.h>
#define IN 1
#define OUT 0
main( )
{ int c, nw , state;
  state=OUT;
  nw=0;
  while ( (c=getchar() ) != EOF)
  {
    if (c==' ' || c=='\n' || c=='\t') state=OUT;
    else if (state==OUT) {
      state=IN;
      ++nw;
    }
  }
}

```

```
printf("\n%d\n",nw);
}
```

Rezultat:

U sledecem novom redu se ispisuje broj reci teksta sa ulaza, pa se predje u novi red. Smatramo da rec je ma koji niz znakova koji ne sadri blanko, tab, new line.

Primer 4 *Demonstracija pretprocesorske direktive #define*

```
#include<stdio.h>
/* Racuna sumu dva broja */
#define sum(a,b) ((a)+(b))

/* Racuna kvadrat broja - pogresna verzija */
#define square_w(a) a*a

/* Racuna kvadrat broja */
#define square(a) ((a)*(a))

/* Racuna minimum tri broja */
#define min(a, b, c) (a)<(b) ? ((a)<(c) ? (a) : (c)) : ((b)<(c) ? (b) : (c))

main()
{
    printf("sum(3,5) = %d\n", sum(3,5));
    printf("square_w(5) = %d\n", square_w(5));
    printf("square_w(3+2) = %d\n", square_w(3+2));
    printf("square(3+2) = %d\n", square(3+2));
    printf("min(1,2,3) = %d\n", min(1,2,3));
    printf("min(1,3,2) = %d\n", min(1,3,2));
    printf("min(2,1,3) = %d\n", min(2,1,3));
    printf("min(2,3,1) = %d\n", min(2,3,1));
    printf("min(3,1,2) = %d\n", min(3,1,2));
    printf("min(3,2,1) = %d\n", min(3,2,1));
}
```

Izlaz iz programa:

```
sum(3,5) = 8
square_w(5) = 25
square_w(3+2) = 11
square(3+2) = 25
min(1,2,3) = 1
min(1,3,2) = 1
min(2,1,3) = 1
min(2,3,1) = 1
min(3,1,2) = 1
min(3,2,1) = 1
```

Primer 5 *Demonstracija pretprocesorske direktive #define*

```
#include <stdio.h>
#define KUBW(a) (a * a * a)
#define KUB(a)  ( (a) * (a) * (a))

main()

{

int b=1;

printf("KUB(%d) = %d\n", 2*b+4, KUBW(2*b+4) );
printf("KUB(%d) = %d\n", 2*b+4, KUB(2*b+4) );

}

Izlaz:
KUB(6) = 22
KUB(6) = 216
```

Primer 6 *Ilustracija beskonačne petlje:*

```
#define forever for(;;);
```

Moguće je definisati makroe sa argumentima tako da tekst zamene bude različit za različita pojavljivanja makroa.

Primer 7

```
#define max(A, B) ((A)>(B) ? (A) : (B))
```

na osnovu ovoga će linija

```
x=max(p+q, r+s)
```

biti zamenjena linijom

```
x=((p+q) > (r+s) ? (p+q) : (r+s));
```

Treba voditi računa o sporednim efektima. Sledeća linija koda prouzrokuje uvećanje vrednosti *i* i *j* za dva.

```
max(i++, j++)
```

Takođe treba voditi računa o zagradama. Sledeći makro prouzrokuje neočekivane rezultate za poziv `square(a+1)`

```
#define square(x) x*x
```

Primer 8

```
#include <stdio.h>
#define max1(x,y) (x>y?x:y)
#define max2(x,y) ((x)>(y)?(x):(y))
#define swapint(x,y) { int z; z=x; x=y; y=z; }
#define swap(t,x,y) { \
    t z; \
```

```
        z=x;  \
        x=y;  \
        y=z;  }

main()
{

    int x=2,y=3;

    printf( "max1(x,y) = %d\n", max1(x,y)  );
    /* max1(x,y) = 3 */

    /* Zamena makroom se ne vrši
    unutar niski pod navodnicima*/
    printf( "max1(x=5,y) = %d\n", max1(x,y)  );
    /* max1(x=5,y) = 3 */

    printf( "max1(x++,y++) = %d\n", max1(x++,y++)  );
    /* max1(x++,y++) = 4 */

    printf( "x = %d, y = %d\n", x, y  );
    /* x = 3, y = 5 */

    swapint(x,y);

    printf( "x = %d, y = %d\n", x, y  );
    /* x = 5, y = 3 */

    swap(int,x,y);
    printf( "x = %d, y = %d\n", x, y  );
    /* x = 3, y = 5 */
}

Izlaz:
max1(x,y) = 3
max1(x=5,y) = 3
max1(x++,y++) = 4
x = 3, y = 5
x = 5, y = 3
x = 3, y = 5
```