

Programiranje II

Beleške sa vežbi

Smer *Informatika*
Matematički fakultet, Beograd

Sana Stojanović

27.03.08.

Sadržaj

1 Dinamička alokacija - primeri sa polinomima	3
-----------------------------------------------	---

1 Dinamička alokacija - primeri sa polinomima

1. Napisati program koji sa standardnog ulaza učitava stepen polinoma, koeficijente polinoma (počev od koeficijenta uz najniži stepen) i vrednost promenljive x i pozivom funkcije

```
int vrednost(int *a, int n, int x)
```

računa vrednost polinoma u toj tački. Funkcija treba da radi u najmanjem broju koraka, što znači da je moramo napisati pomoću Hornerove sheme.

Na primer:

ako unesemo stepena $n = 2$ i polinom $P(x) = 3 + 1 * x^1 + 4 * x^2$ dobijamo vrednosti (za uneto $x = 1$) $P(1) = 8$, (za uneto $x = 2$) $P(2) = 21$.

Napomena: Hornerova shema je način za računanje vrednosti polinoma u datoj tački na sledeći način:

$$P(x) = a_0 + a_1 * x^1 + \dots + a_n * x^n$$

odnosno

$$P(x) = (\dots((a_n * x + a_{n-1} * x^{n-1}) * x + a_{n-2} * x^{n-2}) + \dots) * x + a_0$$

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

```
#include <stdlib.h>
```

```
/* Funkcija koja racuna vrednost polinoma u datoj tacki x  
   primenom Hornerove sheme */
```

```
int vrednost(int *a, int n, int x)
```

```
{
```

```
    int vr = 0;        //Pocetna vrednost sume koju racunamo Hornerovom shemom  
    int i;            //Brojac u petlji
```

```
    /* U petlji redom uzimamo koeficijente polinoma pocev od  
       koeficijenta uz najvisi stepen i uracunavamo ih u tekucu  
       sumu u skladu sa shemom */
```

```
    for(i=n; i>=0; i--)  
        vr = vr*x + a[i];
```

```
    return vr;
```

```
}
```

```
/* Funkcija koja stampa polinom stepena n bez clanova sume koji su  
   jednaki nuli */
```

```
void stampaj_polinom(int *a, int n)
```

```
{
```

```
    int i;
```

```
    if (a[0])
```

```

        printf("%d", a[0]);

    for(i=1; i<=n; i++)
        /* Ispred pozitivnih koeficijenata stampamo znak +... */
        if (a[i] > 0)
            printf(" + %d*x^%d", a[i], i);
        /* ...dok ispred negativnih ne stampamo nista. */
        else if (a[i] < 0)
            printf(" %d*x^%d", a[i], i);
    printf("\n");
}

main()
{
    int *a;
    int n, x;

    printf("Unesite stepen polinoma: ");
    scanf("%d", &n);

    /* Alociramo prostor za koeficijente polinoma. Ako je polinom
       stepena n onda ima n+1 koeficijenata */
    a = (int*)calloc(n+1, sizeof(int));
    if (a==NULL)
        exit(1);

    /* Unosimo koeficijente polinoma */
    printf("Unesite koeficijente polinoma pocevsi od
           koeficijenta uz najnizi stepen: ");
    for(i=0; i<=n; i++)
        scanf("%d", &a[i]);

    /* Unosimo vrednost promenljive */
    printf("Unesite vrednost promenljive x: \n");
    scanf("%d", &x);

    /* Pozivom funkcije racunamo vrednost polinoma u tacki x */
    printf("Vrednost polinoma ");
    stampaj_polinom(a, n);
    printf("u tacki %d je %d.\n", x, vrednost(a, n, x));

    free(a);
}

```

2. Napisati program koji sa standardnog ulaza učitava dva polinoma i računa njihov proizvod. Prvo se unosi stepen prvog polinoma pa njegovi koefici-

jenti (počev od koeficijenta uz najniži stepen) pa zatim stepen drugog polinoma pa njegovi koeficijenti i na standardni izlas se ispisuje polinom koji je jednak njihovom proizvodu. Takođe, pozivom funkcije *vrednost* (iz prethodnog zadatka) proveriti da li je vrednost proizvoda dobro izračunata.

Primer:

za uneto $n = 1$ i polinom $P(x) = 1 - x$, i uneto $m = 1$ i polinom $Q(x) = 1 + x$ rezultujući polinom je $R(x) = 1 - x^2$.

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

/* Funkcija koja stampa polinom stepena n bez clanova sume koji su
   jednaki nuli */
void stampaj_polinom(int *a, int n)
{
    int i;

    if (a[0])
        printf("%d", a[0]);

    for(i=1; i<=n; i++)
        /* Ispred pozitivnih koeficijenata stampamo znak +... */
        if (a[i] > 0)
            printf(" + %d*x^%d", a[i], i);
        /* ...dok ispred negativnih ne stampamo nista. */
        else if (a[i] < 0)
            printf(" %d*x^%d", a[i], i);
    printf("\n");
}

main()
{
    int *a, *b, *c;      /* Pokazivaci na koeficijente polinoma */
    int n, m, p;        /* Stepeni polinoma */
    int i, j;           /* Brojaci u petljama */

    printf("Unesite stepen prvog polinoma: ");
    scanf("%d", &n);

    /* Alociramo prostor za koeficijente prvog polinoma. Ako je polinom
       stepena n onda ima n+1 koeficijenata */
    a = (int*)calloc(n+1, sizeof(int));
    if (a==NULL)
        exit(1);
}
```

```

/* Unosimo koeficijente prvog polinoma */
printf("Unesite koeficijente polinoma pocevsi od
      koeficijenta uz najnizi stepen: ");
for(i=0; i<=n; i++)
    scanf("%d", &a[i]);

printf("Unesite stepen drugog polinoma: ");
scanf("%d", &m);

/* Alociramo prostor za koeficijente drugog polinoma. Ako je polinom
stepena m onda ima m+1 koeficijenata */
a = (int*)calloc(m+1, sizeof(int));
if (a==NULL)
    exit(1);

/* Unosimo koeficijente drugog polinoma */
printf("Unesite koeficijente polinoma pocevsi od
      koeficijenta uz najnizi stepen: ");
for(i=0; i<=m; i++)
    scanf("%d", &b[i]);

/* Kada smo uneli polinome znamo da je stepen rezultujuceg
polinoma (koji je jednak proizvodu prva dva) jednak
zbiru stepena polinoma koji se mnoze pa u skladu sa tim
rezervisemo prostor za n+m+1 koeficijent */
c = (int*)calloc(n+m+1, sizeof(int));

/* Racunamo vrednosti koeficijenta polinoma c. Svi proizvodi
oblika a[i]*b[j] ce biti deo nekog faktora uz x^{i+j}.
Kako polinom c na pocetku alociramo sa calloc, vrednosti
njegovih elemenata ce biti jednake 0 pa sledecim dvostrukom
petljom rasporedjujemo proizvode a[i]*b[j] uz faktor
x^{i+j}. */
for(i=0; i<=n; i++)
    for(j=0; j<=m; j++)
        c[i][j] = c[i][j] + a[i]*b[j];

printf("Proizvod polinoma: \n");
stampaj_polinom(a, n);
stampaj_polinom(b, m);
printf("\n");
stampaj_polinom(c, n+m);

/* Pozivom funkcije vrednost mozemo proveriti da li smo dobro
izracunali proizvod polinoma */

```

```
printf("Unesite vrednost polinoma: ");
scanf("%d", &x);

printf("P(%d)*Q(%d) = %d\n", x, x, vrednost(a,n,x)*vrednost(b,m,x));
printf("R(%d) = %d\n", x, vrednost(c,n+m,x));

free(a);
free(b);
free(c);
}
```