

Programiranje II

Beleške sa vežbi

Smer *Informatika*

Matematički fakultet, Beograd

Sana Stojanović

27.03.08.

Sadržaj

1	Dinamička alokacija - primeri sa polinomima	3
2	Dinamička alokacija - razni zadaci	19

1 Dinamička alokacija - primeri sa polinomima

1. Napisati program koji sa standardnog ulaza učitava prvo stepen polinoma n pa zatim koeficijente dva polinoma tog stepena (počevši od koeficijenta uz najniži stepen), izračunava njihov zbir i štampa ga na standardni izlaz (na standardan način).

Zbir polinoma datih nizom svojih koeficijenata (koji su smešteni u nizove a i b) se dobija sabiranjem elemenata tih nizova po indeksu i koji uzima vrednosti od 0 do n (koeficijenti rezultujućeg polinoma će biti smešteni u nizu koji zadovoljava jednakost: $c[i] = a[i] + b[i]$).

Na primer, ulaz:

```
2
3 1 4
2 5 1
```

predstavlja dva polinoma stepena $n = 2$, $P(x) = 3 + 1 * x^1 + 4 * x^2$ i $Q(x) = 2 + 5 * x^1 + 1 * x^2$ i na standardni izlaz ispisuje njihov zbir u obliku:

```
5 + 6*x^1 + 5*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; /* Pokacivaci na koeficijente polinoma */
    int n;          /* Stepen polinoma */
    int i;          /* Brojac u petljama */

    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);

    b = (int*)calloc(n+1, sizeof(int));
    if (b==NULL)
        exit(1);

    c = (int*)calloc(n+1, sizeof(int));
    if (c==NULL)
        exit(1);

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

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

    /* Racunamo koeficijente zbira polinoma a i b. Zbir polinoma se
       dobija tako sto se sabiraju koeficijenti uz jednake stepene.
       Kako su koeficijenti smesteni u niz ovo odgovara sabiranju
       elemenata nizova a i b po jednakim indeksima i smestanje
       rezultata u niz c. */
    for(i=0; i<=n; i++)
        c[i] = a[i] + b[i];

    /* Stapanje dobijenog polinoma c koji je zbir polinoma a i b */
    printf("Zbir polinoma:\n");
    stampaj_polinom(a, n);
}

```

```

    stampaj_polinom(b, n);
    printf("\n");
    stampaj_polinom(c, n);

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

```

2. Napisati funkciju koja računa zbir polinoma jednakih stepena ako su polinomi zadati nizom svojih koeficijenata i stepenom polinoma.

```

int* zbir(int *a, int *b, int n)

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

/* Funkcija koja racuna zbir dva polinoma a i b predstavljenih nizom
   svojih koeficijenata i stepenom */
int* zbir(int *a, int *b, int n)
{
    int i;
    int *c;

    /* Alociramo prostor za koeficijente zbira unutar funkcije (da
       bi mogli da izracunamo vrednosti tog niza i da ga vratimo
       kao povratnu vrednost funkcije). Kako koristimo dinamiciku
       alokaciju ovo je moguće. */
    c = (int*)calloc(n+1, sizeof(int));
    for(i=0; i<=n; i++)
        c[i] = a[i] + b[i];

    return c;
}

/* 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++)
        if (a[i] > 0)

```

```

        printf(" + %d*x^%d", a[i], i);
    else if (a[i] < 0)
        printf(" %d*x^%d", a[i], i);
    printf("\n");
}

main()
{
    int *a, *b, *c; /* Pokacivaci na koeficijente polinoma */
    int n;          /* Stepen polinoma */
    int i;          /* Brojac u petljama */

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

    /* Ako koristimo funkciju za dobijanje vrednosti polinoma zбира
    prostor za taj niz ne alociramo unutar funkcije main vec
    unutar same funkcije koja racuna zbir */

    /* 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);

    b = (int*)calloc(n+1, sizeof(int));
    if (b==NULL)
        exit(1);

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

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

    /* Racunamo koeficijente zбира polinoma a i b pozivom funkcije
    koja racuna zbir koeficijenata */
    c = zbir(a, b, n);

    /* Stapanje dobijenog polinoma c koji je zbir polinoma a i b */

```

```

printf("Zbir polinoma:\n");
stampaj_polinom(a, n);
stampaj_polinom(b, n);
printf("\n");
stampaj_polinom(c, n);

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

```

3. Napisati program za sabiranje polinoma različitih stepena. Sa standardnog ulaza se učitava prvo stepen prvog polinoma, pa stepen drugog polinoma pa tek nakon toga koeficijenti prvog polinoma pa koeficijenti drugog polinoma. Na standardni izlaz ispisati zbir unesenih polinoma.

Napomena: Funkcija *zbir* (iz prethodnog zadatka) je napisana za sabiranje polinoma jednakih stepena ali čak iako se stepeni polinoma *a* i *b* razlikuju ova funkcija opet može da se iskoristi za računanje zbira tih polinoma pod pretpostavkom da su nizovi *a* i *b* dobijeni pomoću funkcije *calloc* i da je argument *n* koji se prenosi funkciji *zbir* jednak maksimalnom stepenu ta dva polinoma.

```

main()
{
    int *a, n_a; /* Niz u kome su sacuvani koeficijenti polinoma a
                 i stepen tog polinoma. */
    int *b, n_b; /* Niz u kome su sacuvani koeficijenti polinoma b
                 i stepen tog polinoma. */
    int *c, n;   /* Niz u kome cemo cuvati koeficijente polinoma
                 koji ce biti jednak sumi prethodna dva polinoma
                 i njegov stepen */

    int i;      /* Brojac u petljama */

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

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

    /* Stepen zbira polinoma a i b je jednak vecem od stepena
       polinoma a i b. */
    if (n_a > n_b)
        n = n_a;
    else

```

```

        n = n_b;

/* Veci od stepena polinoma a i b ce biti stepen rezultujuceg
   polinoma, ali i same polinome a i b moramo da prosirimo na tu
   dimenziju (zbog funkcije zbir). Ovdje je bitno da polinome
   dobijemo pozivom funkcije calloc da bi bili popunjeni nulama. */

/* Alociranje memorije za koeficijente polinoma a */
a = (int*)calloc(n+1, sizeof(int));
if (a==NULL)
    exit(1);

/* Alociranje memorije za koeficijente polinoma b */
b = (int*)calloc(n+1, sizeof(int));
if (b==NULL)
    exit(1);

/* Prilikom unosenja samih koeficijenata polinoma a i b moramo da
   vodimo racuna da unosimo onoliko koeficijenata koliko je
   predvidjeno stepenom polinoma. */

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

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

/* Pozivom funkcije zbir racunamo zbir polinoma a i b. Kako su
   nizovi a i b dobijeni funkcijom calloc znaci da su popunjeni
   nulama na onim pozicijama gde nema unetih koeficijenata sa
   ulaza (sto ne kviri vrednost polinoma) pa ih mozemo tretirati
   kao polinome jednake duzine (jednake vecoj od duzina polinoma
   a i b) */
c = zbir(a, b, n);

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

```


4. Napisati program koji simulira množenje polinoma skalarom. Sa standardnog ulaza se prvo unosi stepen polinoma pa koeficijenti polinoma počevši od koeficijenta uz najniži stepen pa nakon toga skalar (ceo broj) kojim množimo polinom i ispisuje se rezultujući polinom na standardni izlaz.

*Proizvod polinoma, predstavljenog nizom svojih koeficijenata a , i skalara c se dobija množenjem elemenata niza a tim skalarom. Koeficijenti rezultujućeg polinoma će biti smešteni u niz b koji zadovoljava jednakost $b[i] = c * a[i]$.*

```
#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++)
        if (a[i] > 0)
            printf(" + %d*x^%d", a[i], i);
        else if (a[i] < 0)
            printf(" %d*x^%d", a[i], i);
    printf("\n");
}

main()
{
    int *a, *b;      /* Pokacivaci na koeficijente polinoma */
    int n;           /* Stepen polinoma */
    int i;           /* Brojac u petljama */
    int c;           /* Skalar kojim mnozimo polinom */

    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);

    b = (int*)calloc(n+1, sizeof(int));
    if (b==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]);

    printf("Unesite skalar kojim zelite da pomnozite polinom: ");
    scanf("%d", &c);

    /* Racunamo koeficijente proizvoda polinoma skalarom. Kako skalar ne
       utice na stepen promenljive x vec samo na koeficijente, niz
       koji ce predstavljati ovaj proizvod ce imati koeficijente
       dobijene umnozkom niza a skalarom c. */
    for(i=0; i<=n; i++)
        b[i] = a[i] * c;

    /* Stampanje dobijenog polinoma b koji je jednak proizvodu
       polinoma a skalarom c */
    printf("Proizvod polinoma: \n");
    stampaj_polinom(a, n);
    printf("skalarom %d je: \n", c);
    stampaj_polinom(b, n);

    free(a);
    free(b);
}

```

5. Napisati funkciju koja množi polinom skalarom. Polinom je zadat nizom svojih koeficijenata i stepenom, treći argument funkcije je skalar kojim množimo polinom.

```

int* mnozenje_skalarom(int *a, int n, int c)

#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++)
        if (a[i] > 0)
            printf(" + %d*x^%d", a[i], i);
        else if (a[i] < 0)
            printf(" %d*x^%d", a[i], i);
    printf("\n");
}

/* Funkcija koja mnovi polinom skalarom */
int* mnozenje_skalarom(int *a, int n, int c)
{
    int *b;
    int i;

    b = (int*)calloc(n+1, sizeof(int));
    if (b==NULL)
        exit(1);

    for(i=0; i<=n; i++)
        b[i] = a[i] * c;

    return b;
}

main()
{
    int *a, *b;      /* Pokacivaci na koeficijente polinoma */
    int n;           /* Stepen polinoma */
    int i;           /* Brojac u petljama */
    int c;           /* Skalar kojim mnozimo polinom */

    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]);
}

```

```

printf("Unesite skalar kojim zelite da pomnozite polinom: ");
scanf("%d", &c);

/* Racunamo koeficijente proizvoda polinoma skalarom pozivom
   funkcije */
b = mnozenje_skalarom(a, n, c);

/* Stapanje dobijenog polinoma b koji je jednak proizvodu
   polinoma a skalarom c */
printf("Proizvod polinoma: \n");
stampaj_polinom(a, n);
printf("skalarom %d je: \n", c);
stampaj_polinom(b, n);

free(a);
free(b);
}

```

6. Napisati program koji simulira množenje polinoma promenljivom. Sa ulaza se unosi stepen polinoma i koeficijenti polinoma počevši od koeficijenta uz najniži stepen i na standardni izlaz se štampa rezultujući polinom dobijen množenjem unetog polinoma promenljivom x .

Primer:

Za uneto $n = 2$ i koeficijente 1 2 3 (što predstavlja polinom $P(x) = 1 + 2 * x^1 + 3 * x^2$) rezultujući polinom, dobijen množenjem polinoma $P(x)$ promenljivom x je:

$$P(x) * x = 0 + 1 * x^1 + 2 * x^2 + 3 * x^3$$

Množenje polinoma, predstavljenog nizom svojih koeficijenata a , i promenljive se dobija šiftovanjem elemenata tog niza za jedno mesto u desno. Koeficijenti rezultujućeg polinoma će biti smešteni u niz b koji zadovoljava jednakost: $b[0] = 0$, $b[i] = a[i - 1]$, za $i > 0$.

```

#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;

    /* U ovom zadatku stampamo koeficijent uz x^0 u svakom slucaju da bismo
       videli kako se formira 0 na pocetku rezultujućeg polinoma */

```

```

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

for(i=1; i<=n; i++)
    if (a[i] > 0)
        printf(" + %d*x^%d", a[i], i);
    else if (a[i] < 0)
        printf(" %d*x^%d", a[i], i);
printf("\n");
}

main()
{
    int *a, *b;        /* Pokacivaci na koeficijente polinoma */
    int n;             /* Stepen polinoma */
    int i;             /* Brojac u petljama */

    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);

    /* Stepen rezultujuceg polinoma proizvoda polinoma promenljivom
       je za jedan veci od polaznog polinoma, znaci n+1, pa moramo
       da rezervisemo prostor za n+2 koeficijenta */
    b = (int*)calloc(n+2, sizeof(int));
    if (b==NULL)
        exit(1);

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

    /* Racunamo koeficijente proizvoda polinoma promenljivom.
       Nulti koeficijent novog polinoma bice 0 a svi ostali
       koeficijenti se dobijaju pomeranjem elemenata niza a za jedno
       mesto u desno. */
    b[0] = 0;
    for(i=1; i<=n+1; i++)
        b[i] = a[i-1];

```

```

        /* Stapanje dobijenog polinoma b koji je jednak proizvodu
           polinoma a promenljivom */
        printf("Proizvod polinoma: \n");
        stampaj_polinom(a, n);
        printf("promenljivom x je: \n");
        stampaj_polinom(b, n+1);

        free(a);
        free(b);
    }

```

7. Napisati funkciju koja množi polinom promenljivom. Funkciji prosleđujemo samo koeficijente polinoma i njegov stepen. Funkcija kao povratnu vrednost vraća niz dimenzije za jedan veće od polaznog niza.

```

int* mnozenje_promenljivom(int *a, int n)

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

/* Funkcija koja racuna koeficijente polinoma dobijenog mnozenjem
   polinoma a promenljivom */
int* mnozenje_promenljivom(int *a, int n)
{
    int *b;
    int i;

    /* Stepen rezultujućeg polinoma proizvoda polinoma promenljivom je
       za jedan veći od polaznog polinoma, znači n+1, pa moramo da
       rezervisemo prostor za n+2 koeficijenta */
    b = (int*)calloc(n+2, sizeof(int));
    if (b==NULL)
        exit(1);

    b[0] = 0;
    for(i=1; i<=n+1; i++)
        b[i] = a[i-1];

    return b;
}

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

```

```

/* U ovom zadatku stampamo koeficijent uz x^0 u svakom slucaju da bismo
videli kako se formira 0 na pocetku rezultujuceg polinoma */
printf("%d", a[0]);

for(i=1; i<=n; i++)
    if (a[i] > 0)
        printf(" + %d*x^%d", a[i], i);
    else if (a[i] < 0)
        printf(" %d*x^%d", a[i], i);
printf("\n");
}

main()
{
    int *a, *b;        /* Pokacivaci na koeficijente polinoma */
    int n;            /* Stepen polinoma */
    int i;            /* Brojac u petljama */

    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]);

    /* Racunamo koeficijente proizvoda polinoma promenljivom pozivom
funkcije. Ono \v sto moramo imati u vidu je da je sada niz b
duzine n+2 */
    b = mnozenje_promenljivom(a, n);

    /* Stapanje dobijenog polinoma b koji je jednak proizvodu
polinoma a promenljivom */
    printf("Proizvod polinoma: \n");
    stampaj_polinom(a, n);
    printf("promenljivom x je: \n");
    stampaj_polinom(b, n+1);
}

```

```

    free(a);
    free(b);
}

```

8. Napisati program za izračunavanje koeficijenata polinoma $T_n(x)$ (n je dati prirodan broj koji se učitava sa standardnog ulaza) ako je poznato da važi:

- $T_0(x) = 1$
- $T_1(x) = x$
- $T_n(x) = 2x * T_{n-1}(x) - T_{n-2}(x)$

Napisati i funkciju koja štampa polinom u obliku $T_n(x) = a_0 + a_1 * x^1 + \dots + a_n * x^n$.

Rezultati programa za $n = 2$, $n = 3$ i $n = 4$:

$$T_2 = -1 + 2x^2$$

$$T_3 = -3x^1 + 4x^3$$

$$T_4 = 1 - 8x^2 + 8x^4$$

```

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

/* Funkcija koja stampa polinom stepena n
   - jednostavnija verzija */
void stamp_pol(int *a, int n)
{
    int i;

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

    for(i=1; i<=n; i++)
        printf(" + %d*x^%d", a[i], i);

    printf("\n");
}

/* Funkcija koja stampa polinom stepena n */
void stampaj_polinom(int *a, int n)
{
    int i;

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

    for(i=1; i<=n; i++)
        if (a[i] > 0)

```



```

        printf(" + %d*x^%d", a[i], i);
    else if (a[i] < 0)
        printf(" %d*x^%d", a[i], i);
    printf("\n");
}

main()
{
    int *t_0, *t_1, *t_n, *pom; /* Pokazivaci preko kojih cemo cuvati
                                koeficijente polinoma */

    int n; /* Stepen polinoma koji racunamo */

    int i, j; /* Brojaci u petljama */

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

    /* Kako polinom stepena n ima n+1 koeficijenata, prostor za nizove
       cemo alocirati u skladu sa tim, odnosno rezervisacemo za sve
       nizove prostor velicine n+1 */
    /* Koristimo funkciju calloc da bi nam popunila nizove nulama */
    t_0 = (int *)calloc(n+1, sizeof(int));
    if (t_0 == NULL)
        exit(1);

    t_1 = (int *)calloc(n+1, sizeof(int));
    if (t_1 == NULL)
    {
        free(t_0);
        exit(1);
    }

    t_n = (int *)calloc(n+1, sizeof(int));
    if (t_n == NULL)
    {
        free(t_0);
        free(t_1);
        exit(1);
    }

    /* Odredjujemo koeficijente po postavci zadatka */
    t_0[0] = 1;

    /* Pre nego sto pristupimo t_1[1] proveravamo da li je n>=1 jer se
       moze desiti da je korisnik trazio polinom T_0, odnosno da je n

```

```

    bilo 0. Tada ce za sve nizove biti rezervisan prostor za samo 1
    element. */
if (n>=1)
{
    t_1[0] = 0;
    t_1[1] = 1;
}

printf("\nP_%d(x) = ", n);
if (n == 0)
    stampaj_polinom(t_0, 0);
else if (n == 1)
    stampaj_polinom(t_1, 1);
else
{
    /* Da bismo izracunali koeficijente polinoma T_n moramo
       redom izracunati koeficijente polinoma T_i za i=2,3... */
    for(i=2; i<=n; i++)
    {
        /* Racunamo koeficijente polinoma T_i po rekurzivnoj
           formuli datoj u postavci zadatka*/
        t_n[0] = -t_0[0];

        for(j=1; j<=i; j++)
            t_n[j] = 2*t_1[j-1] - t_0[j];

        /* Podesavamo vrednosti pokazivaca da odgovaraju
           sledecoj iteraciji.
           Prvo cuvamo memorijski prostor na koji je pokazivao
           t_0 (pre dodele t_0=t_1) da ne bi doslo do memory leak-a,
           a i trebace nam kasnije kao prostor za t_n */
        pom = t_0;
        t_0 = t_1;
        t_1 = t_n;
        t_n = pom;
    }

    /* Sada nam se posledji izracunati polinom nalazi u t_1 (nakon
       dodele t_1=t_n, pa njega i stampamo */
    stampaj_polinom(t_1, n);
}

/* Oslobadjamo memoriju */
free(t_0);
free(t_1);
free(t_n);

```

```
}
```

2 Dinamička alokacija - razni zadaci

1. Sastaviti program koji omogućava korisniku da unosi sa tastature niz celih pozitivnih ili negativnih brojeva kojih nema više od n (n se unosi sa standardnog ulaza). Za svaki uneti broj, negativne vrednosti se upisuju na uzastopne pozicije počev od početka niza, a pozitivne vrednosti na uzastopne pozicije počev od kraja niza. Unos se završava kada se unese 0 ili kada je niz popunjen. Kada je unos završen, program ispisuje sadržaj popunjenog dela niza.

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

main() {
    int *a;      /* Pokazivac na niz celih brojeva */
    int n;      /* Dimenzija niza */

    int i, j;   /* Brojaci koje cemo koristiti u petljama */
    int br;     /* Pomocna promenljiva u koju cemo ucitavati brojeve
                sa ulaza */

    printf("Unesite broj n: \n");
    scanf("%d", &n);

    /* Pozivamo funkciju calloc da bismo niz popunili nulama u startu.
       Kako nam je 0 oznaka za kraj unosa (odnosno 0 sigurno nece biti
       element naseg niza), na taj nacin cemo znati koji su elementi
       niza popunjeni */
    a = (int *)calloc(n, sizeof(int));
    if (a == NULL)
        exit(1);

    printf("Unesite cele brojeve (0 za kraj):\n");

    /* i ce nam predstavljati popunjenost negativnim brojevima
       koje smestamo na pocetak niza i odgovarace indeksu poslednjeg
       negativnog broja u nizu, i ce se povecavati u toku programa*/

    /* j ce nam predstavljati popunjenost pozitivnim brojevima koje
       smestamo na kraj niza i odgovarace indeksu poslednjeg pozitivnog
       broja u nizu, j ce se smanjivati u toku programa */

    /* Uslov i<=j nam sluzi da bismo iz petlje izašli u trenutku kada
```

```

        u nizu nema vise mesta */
/* obratite paznju: ne smemo staviti
   for(i=0, j=n-1; i<=j; i++, j--) */

for(i=0, j=n-1; i <= j;)
{
    /* Citamo novi broj sa standardnog ulaza */
    scanf("%d", &br);

    /* Ako smo uneli 0 prekidamo */
    if (!br) break;

    /* Ako je negativan broj smestamo ga na pocetak niza,
       a ako je pozitivan smestamo ga na kraj.
       Tom prilikom se odgovarajuci brojac pomera na sledece mesto
       u nizu (odnosno brojac negativnih za jedno mesto u desno, a
       brojac pozitivnih za jedno mesto u levo). */
    if (br<0)
        a[i++] = br;
    else
        a[j--] = br;
}

/* Ispis popunjenog dela niza. Stampamo samo one elemente koji se
   razlikuju od 0 */
printf("Uneli ste sledece elemente: \n");
for(i=0; i<n; i++)
    if (a[i])
        printf("%d ", a[i]);

    free(a);
}

```

2. Napisati program koji za datoteku čije se ime zadaje kao prvi argument komandne linije ispisuje podatak o dužinama svih reči koje se pojavljuju u toj datoteci i broju reči te dužine.

Na primer, za datoteku:

```
aaa aa aaa aa aaa aaa
```

Reč dužine 3 se pojavila 4 puta a reč dužine 2 se pojavila 2 puta.

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

```

#define KORAK 10

typedef struct word{
    int duz;      /* Duzina reci koja se pojavila u datoteci */
    int br;      /* Broj pojavljivanja reci date duzine */
} word;

main()
{
    word *a = NULL;      /* Pokazivac na niz reci */
    word *a_pom;        /* Pomocni pokazivac */
    int duzina = 0, alocirano = 0;
    int length;

    int i, max_i;
    char s[100];        /* Pomocna promenljiva u kojoj cuvamo
                        tekucu rec */

    FILE *in;

    in=fopen(argv[1], "r");

    if (in==NULL)
    {
        fprintf(stderr, "\nGreska pri otkrivanju datoteke: %s\n", argv[1]);
        exit(1);
    }

    /* Petlja koja nam sluzi da napravimo niz svih reci koje se pojavljuju
    u datoteci. Osim reci koja je procitana u nizu cuvamo i broj
    pojavljivanja te reci. */
    do
    {
        if (duzina == alocirano)
        {
            /* Niz se prosiruje za 10 elemenata vise */
            alocirano = alocirano + KORAK;

            a_pom = (word *) realloc(a, alocirano*sizeof(word));

            if (a_pom != NULL)
                a = a_pom;
            else
            {
                free(a);
            }
        }
    }

```

```

        exit 1;
    }

}

/* Citamo tekucu rec */
fscanf(in, "%s", s);

/* Proveravamo da li se duzina te reci ranije pojavila u
   datoteci i ako jeste uvecavamo njen broj pojavljivanja za 1 */
length = strlen(s);

for(i=0; i<duzina; i++)
    if (length == a[i].duz)
        {
            a[i].br++;
            break;
        }

/* Ako se rec te duzine nije ranije pojavila onda treba da
   unesemo rec te duzine u niz. Kako prethodna petlja
   proverava da li se duzina nalazi u nizu, u slucaju da se
   ne nalazi vazice i=duzina. */
if (i == duzina)
    {
        a[duzina].duz = length;
        a[duzina].br = 1;
        duzina++;
    }
} while (!feof(in));

/* Ispisujemo podatke o duzinama reci i broju njihovog pojavljivanja */
for(i=0; i < duzina; i++)
    printf("%d\t%d\n", a[i].duz, a[i].br);
}

```