

Polinomi – algoritmi za množenje i FFT

Napisati malu biblioteku za rad sa polinomima(najviše stepena 20):

1. definisati strukturu POLINOM koja predstavlja polinom i sadrži polja stepen i koeficijenti
2. napisati funkciju koja ispisuje polinom na standardni izlaz
3. napisati funkciju koja učitava polinom sa standardnog ulaza
4. napisati funkciju koja računa vrednost polinoma u nekoj tački korišćenjem Hornerove sheme
5. napisati funkciju koja računa zbir polinoma
6. napisati funkciju koja računa proizvod polinoma

=====
kod razdvojiti na polinom.c, test_polinom.c i polinom.h

format polinom.h datoteke

```
#ifndef POLINOM_H
#define POLINOM_H 1
```

.....

```
#endif
```

pokretanje:

```
gcc -o pokreni polinom.c test_polinom.c
```

ili preko objektnih fajlova:

```
gcc -c polinom.c
gcc -c test_polinom.c
gcc -o pokreni polinom.o test_polinom.o
```

ili automatizovano:

makefile:

```
pokreni: polinom.o test_polinom.o
gcc -o pokreni polinom.o test_polinom.o
```

```
polinom.o: polinom.c polinom.h
gcc -c polinom.c
```

```
test_polinom.o: test_polinom.c polinom.h
gcc -c test_polinom.c
```

u komandnoj liniji kucamo samo make

polinom.h

```
#ifndef _POLINOM_H
#define _POLINOM_H
```

```
/* Maksimalni stepen polinoma */
```

```
#define MAKS_STEPEN 20
```

```
/* Polinomi se predstavljaju strukturom koja cuva koeficijente
   (koef[i] je koeficijent uz clan x^i) i stepen polinoma */
typedef struct {
    double koef[MAKS_STEPEN + 1];
    int stepen;
} Polinom;

/* Funkcija koja ispisuje polinom na standardni izlaz u citljivom
   obliku. Polinom se prenosi po adresi da bi se uštedela memorija:
   ne kopira se cela struktura, vec se samo prenosi adresa na kojoj
   se nalazi polinom koji ispisujemo */
void ispisi(const Polinom * p);

/* Funkcija koja ucitava polinom sa standardnog ulaza */
Polinom ucitaj();

/* Funkcija racuna i vraca vrednost polinoma p u tacki x Hornerovim
   algoritmom */
double izracunaj(const Polinom * p, double x);

/* Funkcija koja sabira dva polinoma */
Polinom saberi(const Polinom * p, const Polinom * q);

/* Funkcija koja mnozi dva polinoma p i q */
Polinom pomnozi(const Polinom * p, const Polinom * q);

#endif
```

polinom.c

```
#include <stdio.h>

#include <stdlib.h>

#include "polinom.h"

void ispisi(const Polinom * p)
{
    int nulaPolinom = 1;
    int i;
    /* Ispisivanje polinoma pocinje od najviseg stepena ka najnižem
    da bi polinom bio ispisan na prirodan nacin. Ispisuju se samo
    oni koeficijenti koji su razliciti od nule. Ispred pozitivnih
    koeficijenata je potrebno ispisati znak + (osim u slucaju
    koeficijenta uz najvisi stepen). */
    for (i = p->stepen; i >= 0; i--) {

        if (p->koef[i] {

            /* Polinom nije nula polinom, cim je neki od koeficijenata
            razlicit od nule */
            nulaPolinom = 0;
            if (p->koef[i] >= 0 && i != p->stepen)
                putchar('+');
            if (i > 1)
                printf("%.2fx^%d", p->koef[i], i);
            else if (i == 1)
                printf("%.2fx", p->koef[i]);
```

```

else
    printf("%.2f", p->koef[i]);
}
}
/* U slucaju nula polinoma indikator ce imati vrednost 1 i tada
se ispisuje nula. */
if (nulaPolinom)
    printf("0");
    putchar('\n');
}

Polinom ucitaj()
{
    int i;
    Polinom p;

    printf("Unesite polinom p (prvo stepen, pa zatim koeficijente");
    printf(" od najveceg stepena do nultog):\n");

    /* Ucitava se stepena polinoma */
    scanf("%d", &p.stepen);

    /* Ponavlja se ucitavanje stepena sve dok se ne unese stepen iz
    dozvoljenog opsega */
    while (p.stepen > MAKS_STEPEN || p.stepen < 0) {
        printf("Stepen polinoma pogresno unet, pokusajte ponovo: ");
        scanf("%d", &p.stepen);
    }
}

```

```

}

/* Unose se koeficijenti polinoma */
for (i = p.stepen; i >= 0; i--)
    scanf("%lf", &p.koef[i]);

/* Vraca se procitani polinom */
return p;
}

double izracunaj(const Polinom * p, double x)
{
    /* Primer: Hornerov algoritam za polinom x^4+2x^3+3x^2+2x+1:
       x^4+2x^3+3x^2+2x+1 = (((x+2)*x + 3)*x + 2)*x + 1 */

    double rezultat = 0;
    int i = p->stepen;

    /* Rezultat se na pocetku inicijalizuje na nulu, a potom se u
       svakoj iteraciji najpre mnozi sa x, a potom i uvecava za
       vrednost odgovarajuceg koeficijenta */
    for (; i >= 0; i--)
        rezultat = rezultat * x + p->koef[i];
    return rezultat;
}

```

```

Polinom saberi(const Polinom * p, const Polinom * q)

```

```

{
    Polinom r;
    int i;

    /* Stepen rezultata ce odgovarati stepenu polinoma sa vecim
       stepenom */
    r.stepen = p->stepen > q->stepen ? p->stepen : q->stepen;

    /* Racunaju se svi koeficijenti rezultujuceg polinoma tako sto se
       sabiraju koeficijenti na odgovarajucim pozicijama polinoma koje
       sabiramo. Ukoliko je pozicija za koju se racuna koeficijent veca
       od stepena nekog od polaznih polinoma podrazumeva se da je
       koeficijent jednak koeficijentu uz odgovarajuci stepen iz drugog
       polinoma */
    for (i = 0; i <= r.stepen; i++)
        r.koef[i] =
            (i > p->stepen ? 0 : p->koef[i]) +
            (i > q->stepen ? 0 : q->koef[i]);

    /* Vraca se izracunati polinom */
    return r;
}

```

Polinom pomnozi(const Polinom * p, const Polinom * q)

```

{
    int i, j;
    Polinom r;

```

```

/* Stepen rezultata odgovara zbiru stepena polaznih polinoma */
r.stepen = p->stepen + q->stepen;
if (r.stepen > MAKS_STEPEN) {
    fprintf(stderr, "Stepen proizvoda polinoma izlazi iz opsega\n");
    exit(EXIT_FAILURE);
}

/* Svi koeficijenti rezultujucega polinoma se inicijalizuju na 0 */
for (i = 0; i <= r.stepen; i++)
    r.koef[i] = 0;

/* U svakoj iteraciji odgovarajuci koeficijent rezultata se
   uvecava za proizvod odgovarajucih koeficijenata iz polaznih
   polinoma */
for (i = 0; i <= p->stepen; i++)
    for (j = 0; j <= q->stepen; j++)
        r.koef[i + j] += p->koef[i] * q->koef[j];

/* Vraca se izracunati polinom */
return r;
}

```

```

Polinom izvod(const Polinom * p)

```

```

{
    int i;
    Polinom r;

```

```

/* Izvod polinoma ce imati stepen za jedan stepen manji od
stepena polaznog polinoma. Ukoliko je stepen polinoma p vec
nula, onda je rezultujuci polinom nula (izvod od konstante je
nula). */
if (p->stepen > 0) {
    r.stepen = p->stepen - 1;

    /* Racunanje koeficijenata rezultata na osnovu koeficijenata
    polaznog polinoma */
    for (i = 0; i <= r.stepen; i++)
        r.koef[i] = (i + 1) * p->koef[i + 1];
} else
    r.koef[0] = r.stepen = 0;

/* Vraca se izracunati polinom */
return r;
}

```

```

Polinom n_izvod(const Polinom * p, int n)

```

```

{

```

```

    int i;

```

```

    Polinom r;

```

```

/* Provera da li je n nenegativna vrednost */

```

```

if (n < 0) {

```

```

    fprintf(stderr, "U n-tom izvodu polinoma, n mora biti >=0 \n");

```



```
    exit(EXIT_FAILURE);
}

/* Nulti izvod je bas taj polinom */
if (n == 0)
    return *p;

/* Za n>=1, n-ti izvod se racuna tako sto se n puta pozove
   funkcija za racunanje prvog izvoda polinoma */
r = izvod(p);
for (i = 1; i < n; i++)
    r = izvod(&r);

/* Vraca se dobijeni polinom */
return r;
}
```

test-polinoma.c

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

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

```
#include "polinom.h"
```

```
int main(int argc, char **argv)
```

```
{
```

```
    Polinom p, q, z, r;
```

```
    double x;
```

```
    int n;
```

```
/* Unosi se polinom p */
```

```
p = ucitaj();
```

```
/* Ispisuje se polinom p */
```

```
ispisi(&p);
```

```
/* Unosi se polinom q */
```

```
q = ucitaj();
```

```
/* Polinomi se sabiraju i ispisuje se izracunati zbir */
```

```
z = saberi(&p, &q);
```

```
printf("Zbir polinoma je polinom z:\n");
```

```
ispisi(&z);
```

```
/* Polinomi se mnoze i ispisuje se izracunati proizvod */
```

```
r = pomnozi(&p, &q);
```

```
printf("Prozvod polinoma je polinom r:\n");
```

```
ispisi(&r);
```

```
/* Ispisuje se vrednost polinoma u unetoj tacki */
```

```
printf("Unesite tacku u kojoj racunate vrednost polinoma z:\n");
```

```
scanf("%lf", &x);
```

```
printf("Vrednost polinoma z u tacki %.2f je %.2f\n", x,
```

```
    izracunaj(&z, x));
```

```
/* Racuna se n-ti izvod polinoma i ispisuje se rezultat */
```

```
printf("Unesite izvod polinoma koji zelite:\n");
```

```
scanf("%d", &n);  
  
r = n_izvod(&r, n);  
  
printf("%d. izvod polinoma r je: ", n);  
  
ispisi(&r);  
  
  
exit(EXIT_SUCCESS);  
  
}
```