JMBAG

IME I PREZIME

Programiranje 1 – 1. ispit, 7. 2. 2025.

Upute: Na ispitus je dozvoljeno koristiti samo pribor za pisanje i brisanje, te službeni podsjetnik. Kalkulatori, razne neslužbene tablice, papiri i sl., nisu dozvoljeni! **Mobitele isključite i spremite!** Sva rješenja napišite isključivo na papire sa zadacima, jer jedino njih predajete. Obavezno predajte sve papire sa zadacima, čak i ako neke zadatke niste rješavali. Ne zaboravite se **potpisati** na svim papirima! Skice smijete raditi i na drugim papirima koje će vam dati dežurni asistent. Zadaci $3-5$ su programski s obzirom na uvjet polaganja.

U svim zadacima zabranjeno je korištenje dodatnih nizova i standardne matematičke biblioteke (zaglavlje math.h), osim ako je u zadatku drugačije navedeno.

Zadatak 1 (4+3+5+6=18 bodova)

(a) Odredite sve baze $b \geq 2$ i uvjete na varijable x, y, takve da vrijedi $2 \cdot (xy)_b > (yx)_{b+1}$.

(b) Izračunajte bez pretvaranja u bazu 10: $((JJJ)_{(20)} - (999)_{(20)}) \cdot (21)_{20}$.

(c) Neka je x 3-bitni prirodni broj. Napravite logički sklop koji računa je li zadani broj x dijeljiv s 3. **Obavezno napišite cijeli postupak**.

(d) Unutar teksta se nalaze imena, prezimena korisnika (počinju velikim slovom) i korisnička imena zadana na sljedeći način: `Ime Prezime - xyBroj`, gdje x predstavlja prva 3 slova imena, y prva tri slova prezimena, a $Broj$ je niz od 0 do maksimalno 3 znamenke. Npr. za korisnika Pero Peric, tražena konstrukcija unutar teksta može biti PerPer ili PerPer3 ili PerPer456 (velika slova imena i prezimena ostaju velika i u korisničkom imenu).

a) Napišite regularni izraz koji prepoznaje ime, prezime i korisničko ime zadano u gore opisanom formatu u proizvoljnom tekstu.

b) Napišite regularni izraz koji prepoznaje niz imena, prezimena i korisničkih imena odvojenih zarezom. Npr. prepoznaje `Pero Peric - PerPer, Mirko Miric - MirMir34, Damir Misko - DamMis742`

RJEŠENJE:

a) $2 \cdot (x \cdot b + y) > y \cdot (b+1) + x \Rightarrow 2xb + 2y > yb + y + x \Rightarrow 2xb - yb > y + x - 2y \Rightarrow b(2x-y) > x - y$
$b > \frac{x-y}{2x-y},\ x, y < b, 2x - y > 0 \Rightarrow b > 1 - \frac{x}{2x-y}, x > y/2$

b) $(JJJ)_{20} - (999)_{20} = (AAA)_{20}$
$(AAA)_{20} * (21)_{20} = (11BAA)_{20}$

c) Tablica istinitosti:

z	x	y	f
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

DNF: $(\bar{z} \cdot \bar{y} \cdot \bar{x}) + \bar{z} \cdot y \cdot x + z \cdot y \cdot \bar{x}$

d) a) `([A-Z][a-z][a-z])[a-z]* ([A-Z][a-z][a-z])[a-z]* - \1\2\d{0,3}`

b) `(([A-Z][a-z][a-z])[a-z]* ([A-Z][a-z][a-z])[a-z]* - \1\2\d{0,3}(, )?)+`

JMBAG

IME I PREZIME

Programiranje 1 – 1. ispit, 7. 2. 2025.

Zadatak 2 (3+6+6 = 15 bodova)

a) Ako želimo spremiti cijeli broj na veličinu jednog byta tipično se koristi zaglavlje `stdint.h` gdje postoji tip `int8_t`. U slučaju da nemate zaglavlje `stdint.h` kako biste kreirali gornji tip imena `INT8_T` koristeći naredbu `typedef`

b) Inicijaliziramo tri varijable `a,b,c` tipa `INT8_T`
`INT8_T` a=1;
`INT8_T` b=4;
`INT8_T` c=127;
Što vraća `sizeof(c)`?

Napišite redom ispis operacija:

```
prinft("%d",a+b);
```

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

```
printf("%d",a&b); /* logicko i bit po bit */
```

```
c=c+a;
prinft("%d",c);
```

```
c=a<<1; /* pomak bitova u lijevo */
prinft("%d",c);
```

c) Napišite funkciju koja prima cjelobrojni tip `INT8_T` i ispisuje prikaz tog broja u računalu kao niz bitova.

RJEŠENJE:

a) `typedef signed char INT8_T;`

b) $1, 5, 1, -128, 2$

c)
```
void ispis(INT8_T broj){
    INT8_T bit,i;
    unsigned char mask=1<<7;
    for (i=0;i<8;++i){
        bit = (broj & mask) ? 1:0;
        printf("%d",bit);
        mask=mask>>1;
    }
}
```

JMBAG

IME I PREZIME

Programiranje 1 – 1. ispit, 7. 2. 2025.

Zadatak 3 (12+8 = 20 bodova)

Za polinom $p(x) = \sum_{i=0}^{n} a_i x^i$ definiramo njegovu k-tu derivaciju kao polinom

$$p^{(k)}(x) = \sum_{i=k}^{n} \frac{i!}{(i-k)!} a_i x^{i-k}.$$

Ako u točki $c \in \mathbb{R}$ polinom p ima ekstrem, onda je $p'(c) = 0$. Više derivacije nam daju dovoljan uvjet za određivanje je li to točka minimuma ili točka maksimuma. Za polinome stupnja barem dva postoji $m \in \mathbb{N}$, takav da za svaki $k \in \{1, \ldots, m\}$ vrijedi $p^{(k)}(c) = 0$ i $p^{(m+1)}(c) \neq 0$, te tada u slučaju

- kada je $p^{(m+1)}(x) > 0$, p ima u x strogi lokalni minimum,
- kada je $p^{(m+1)}(x) < 0$, p ima u x strogi lokalni maksimum.

a) Napišite funkciju `int ekstrem(double a[], int n, double c)` koja prima cijeli broj $n \geq 2$, niz koeficijenata $a_0, a_1, \ldots, a_n$ ($a_n \neq 0$) polinoma p te realan broj c. Funkcija za točku lokalnog ekstrema c polinoma p (ne treba provjeravati) evaluacijom viših derivacija Hornerovim algoritmom treba odrediti vrstu ekstrema te vratiti 0 ako je c strogi lokalni minimum, inače 1.

b) Napišite funkciju `int minimumi(double a[], int n, double x[], int l)` koja prima cijeli broj $n \geq 2$, niz koeficijenata $a_0, a_1, \ldots, a_n$ polinoma p, te niz točaka ekstrema $x_0, x_1, \ldots, x_{l-1}$ polinoma p duljine l. Funkcija treba izbrisati iz niza sve točke maksimuma tako da ostanu samo točke minimuma te vratiti novu duljinu niza. Smijete koristiti funkciju `ekstrem` iz (a) dijela zadatka čak i ako ju niste napisali.

Napomena. Dozvoljeno je korištenje dodatnih (pomoćnih) funkcija. Zabranjeno je korištenje dodatnih nizova.

RJEŠENJE:

```
#include <stdio.h>

//k-tu derivaciju mozemo pogodnije zapisati kao p_k(x) = sum_{j=0}^{n-k} ( (j+k)!/j! * a[j+k] * x^j )

double deriv(double a[], int n, double x, int k); // računa p_k(x)
int fakt(int j, int k); // računa (j+k)!/j!
int ekstrem(double a[], int n, double c);
int minimumi(double a[], int n, double x[], int l);
int minmax(double a[], int n, double x[], int l);

int main(void){
    int i;

    double p[] = {0, 60, 0, -25, 0, 3}; //min: -1, 2  max: 1, -2

    double x[]={2, 2, 1, -2, -1, 2};
    for(i=0; i<5; i++) printf ("%.2f, ", x[i]);
    printf("\nNakon: ");
    int l=minimumi(p, 5, x, 5);
    for(i=0; i<l; i++) printf ("%.2f, ", x[i]);
    printf("\n");


    //alternativni:
    // p(0.5(x+y))= 0 za x=2, y=-2 ili x=1, y=-1
    //p(0.5(x+y))=+-28.4 za x=+-1, y=+-2
    double y[]={-2, 2, 2, 1, -1, -2, -1, 2};
    for(i=0; i<5; i++) printf ("%.2f, ", y[i]);
```

```
    printf("\nNakon: ");
    int l1=minmax(p, 5, y, 5);
    for(i=0; i<l1; i++) printf ("%.2f, ", y[i]);
}


int fakt(int j, int k){
    int res=1;
    while(k>0){
        res*=(j+k);
        k--;
    }
    return res;
}

double deriv(double a[], int n, double x, int k){
    int j;
    double p=0;
    for(j=n-k; j>=0; j--){
        p=x*p+fakt(j,k)*a[j+k];
    }
    return p;
}

int ekstrem(double a[], int n, double c){
    int k;
    for(k=2; k<=n; k++){
        double der=deriv(a,n,c,k);
        if(der){
            if(der>0) return 0;
            else return 1;
        }
    }
}

int minimumi(double a[], int n, double x[], int l){
    int i, j=0, br_obrisanih;
    for(i=0; i<l; ++i){
        if(ekstrem(a, n, x[i])){
            ++br_obrisanih;
            continue;
        }
        if(i>j) x[j]=x[i];
        ++j;
    }
    return l-br_obrisanih;
}

int minmax(double a[], int n, double x[], int l){
    int i, j, br_dodanih=0;
    for(i=1; i<l; ++i){
        if( (ekstrem(a, n, x[i-1]) + ekstrem(a, n, x[i])) == 1) ++br_dodanih;
    }
    int l2=l+br_dodanih;
    j=l2-1;
    for(i=l-1; i>0; --i){
        x[j]=x[i];
        --j;
        if( (ekstrem(a, n, x[i-1]) + ekstrem(a, n, x[i])) == 1 ){
            x[j]=deriv(a, n, 0.5*( x[i-1] + x[i] ), 0);
            --j;
        }
    }
    return l2;
}
```

JMBAG

IME I PREZIME

Programiranje 1 – 1. ispit, 7. 2. 2025.

Zadatak 4 (14 + 13 = 27 bodova) Slavko je vlasnik zalagaonice, i ima $n \leq 200$ artikala koji su kodirani sa jedinstvenim znakom i zapisani u niz **char** $A[\,]$ a njihove cijene su zapisane u nizu **double** $C[\,]$ na način da je $C[i]$ cijena od artikla $A[i]$.

redni broj	Artikl	Cijena/€
0	a	2.50
1	E	55.99
2	h	12.20
3	Q	11.00

(a) Kako bi Slavko imao pregledniju zalagaonicu, napišite funkciju koja sortira artikle uzlazno po cijeni koristeći algoritam sortiranja umetanjem (*Insertion sort*). Ukoliko podzadatak točno riješite nekim drugim algoritmom sortiranja, vrednovat će se sa 7 bodova.

(b) Pod pretpostavkom da je niz artikala sortiran uzlazno po cijeni, napišite funkciju

char Ponuda(**char** Artikl[ ], **double** Cijena[ ], **int** n, **double** N)

koja, koristeći binarno pretraživanje, vraća znak najskupljeg artikla koju kupac može priuštiti količinom novca N.

RJEŠENJE:

```
#include<stdio.h>

//a) Sortiranje po cijeni koristeći algoritam sortiranja umetanjem , insertion sort.

void insertion_sort_po_cijeni( char A[], double C[], int n){
    double tempC;
    char tempA;
    int j;
    for( int i = 1 ; i < n ; i++ ){
        if( C[i] < C[i-1] ){
            tempC = C[i];
            tempA = A[i];
            j = i;
            do{
                C[j] = C[j-1];
                A[j] = A[j-1];
                j--;
            }while( j >=1 && C[j-1] > tempC);
            C[j] = tempC;
            A[j] = tempA;
        }
    }
}

//b) Binarno pretraživanje za pronalazak najskupljeg artikla koji se može kupiti s količinom novca N

char Ponuda( char A[], double C[] , int n , double N){
    int l = 0, d = n-1;
    int s;

    while( l <= d ){

        s = ( l + d )/2;

        if( N < C[s] ){

            d = s - 1;
```

```
        }
        else if( N > C[s] ){

            if( N < C[s+1] ){
                return A[s];
            }
            else{
                l = s + 1;
            }

        }
        else{ //if N == C[s]
            return A[s];
        }

    }
}

// pomnoćna funkcija za printanje tablice ( redni broj : artikl | cijena )
void printNizovi( char Artikl[], double Cijena[], int n ){
    for( int i = 0 ; i < n ; i++ )printf( "%d : %c | %g \n",i, Artikl[i], Cijena[i]);
}

//primjer izvršavanja funkcija iz podzadataka a) i b)

int main(void){
    int n = 10;
    char Artikl[10] = { 'a', 'b', '+', 'd', '1', '(', '=' ,'Q', '!', '¸'};
    double Cijena[10] = { 12, 1.1, 2.33, 130.23, 122.1, 122.01, 122.01, 90, 0.05, 100.001};

    printNizovi(Artikl,Cijena,n);
    insertion_sort_po_cijeni( Artikl, Cijena , n);
    printf("\n");
    printNizovi(Artikl,Cijena,n);

    char x = Ponuda(Artikl, Cijena , n, 122.05 );
    printf( "Nudimo artikl %c", x );

    return 0;
}
```

JMBAG

IME I PREZIME

Programiranje 1 – 1. ispit, 7. 2. 2025.

Zadatak 5 (10 + 10 = 20 bodova)

a) Napišite funkciju `int najcesca_znamenka(int n, int b, int *broj_pojavljivanja)` koja prima prirodne brojeve n i b ($b \geq 2$, ne treba provjeravati) i vraća onu znamenku koja se u zapisu broja n u bazi b javlja najviše puta. Ako takvih ima više, treba vratiti najveću. Kroz varijabilni argument, funkcija treba vratiti broj pojavljivanja najčešće znamenke.

b) Napišite funkciju `int nadi_bazu(int n, int k)` koja prima prirodne brojeve n i k te računa najmanju bazu $b \geq 2$ takvu da zapis broja n u bazi b ima k znamenaka. Ako takva ne postoji, neka funkcija vraća -1.

RJEŠENJE:

```
#include<stdio.h>

int najcesca_znamenka(int n, int b, int *broj_pojavljivanja){
    int max_br=0, max_zn=0, kopija;
    kopija = n;
    while(kopija>0){
        if(kopija%b == 0) max_br++;
        kopija/=b;
    }
    for(int i = 1; i < b; i++){
        kopija = n;
        int broj_trenutne = 0;
        while(kopija>0){
            if(kopija%b == i) broj_trenutne++;
            kopija/=b;
        }
        if(broj_trenutne >= max_br){
            max_zn = i;
            *broj_pojavljivanja = broj_trenutne;
            max_br = broj_trenutne;
        }
    }
    return max_zn;
}

int nadi_bazu(int n, int k){
    int kopija, broj_znamenki, b=2;
    while(b <= n+1){
        broj_znamenki=0;
        kopija = n;
        while(kopija>0){
            kopija/=b;
            broj_znamenki++;
        }
        if(broj_znamenki == k){
            return b;
        }
        else if(broj_znamenki < k){
            return -1;
        }
        b++;
    }
    return -1;
}
```

```
int main(void){
    int pomocni;
    printf("%d\n", nadi_bazu(8,1));
    int rez = najcesca_znamenka(11223345,10,&pomocni);
    printf("%d, %d\n",rez, pomocni);
    return 0;
}
```