

# Programiranje 1 – ispit, 22. 2. 2024.

## Rješenja zadatka

**Zadatak 1** (4+3+4+4=15 bodova)

- (a) Neka je  $b \geq 2$  prirodan broj. Dokažite da ne postoje ne-nul znamenke  $x, y$  u bazi  $b$  takve da je  $(xxx)_{(b)} = (yy)_{(b+1)}$ .
- (b) Izračunajte bez pretvaranja u bazu 10:  $(AAA)_{(13)} \cdot (13)_{(13)}$ .
- (c) Dan je logički izraz  $g = (((a + \bar{b}) \cdot b + \bar{c}) \cdot c + \bar{d}) \cdot d$ .
- (c1) Pojednostavite  $g$  tako da pojednostavljeni izraz ima samo 3 operatora.
- (c2) Zapišite  $g$  bez korištenja operatora konjunkcije.
- (d) U nekom tekstnom dokumentu zapisan je niz pozitivnih decimalnih brojeva. Zapis je formatiran tako da je unutar zagrada navedeno nekoliko decimalnih brojeva (barem jedan), a brojevi su međusobno odvojeni zarezom i razmakom. Najlijevija znamenka brojeva nije 0, osim ako su brojevi između 0 i 1. Primjer zapisa jednog niza je sljedeći:

(0.1, 2.33, 77.0.1, 11.00001)

Napišite regularan izraz koji prepoznaje ovako zapisan niz decimalnih brojeva.

### Rješenja:

- (a) Imamo  $x(b^2 + b + 1) = y(b + 2)$  i  $0 < x, y < b$ . Onda je  $2y - x = b(bx - y + x)$ . Vidimo da je desna strana veća ili jednaka  $2b$  jer je  $bx > y$  pa je  $bx - y + x \geq 2$ . Zaključujemo  $2y - x \geq b$ , ali to je nemoguće jer je  $y < b$ .
- (b) Odgovor je  $(10434)_{(13)}$ .
- (c) Imamo  $(x + \bar{y}) \cdot y = x \cdot y$ , pa korištenjem tog pravila više puta dobivamo

$$\begin{aligned} g &= (((a + \bar{b}) \cdot b + \bar{c}) \cdot c + \bar{d}) \cdot d \\ &= ((a \cdot b + \bar{c}) \cdot c + \bar{d}) \cdot d \\ &= (a \cdot b \cdot c + \bar{d}) \cdot d \\ &= a \cdot b \cdot c \cdot d. \end{aligned}$$

Zapis bez konjunkcije bi bio  $\overline{\bar{a} + \bar{b} + \bar{c} + \bar{d}}$ .

- (d)  $\backslash(((0\backslash.\backslash d+)|([1-9]\backslash d*\backslash.\backslash d+))\backslash, \_)* ((0\backslash.\backslash d+)|([1-9]\backslash d*\backslash.\backslash d+))\backslash)$

# Programiranje 1 – ispit, 22. 2. 2024.

## Rješenja zadataka

### Zadatak 2 (5+6+4=15 bodova)

- (a) Napišite glavni program u kojem se deklariraju dvije varijable  $a$  i  $b$  cjelobrojnog tipa i njihove vrijednosti se inicijaliziraju na 5 i 8. Deklarirajte polje  $c$  od 4 elementa i postavite ih redom na  $2a + 2b$ ,  $a - 2b$ ,  $a * b$ ,  $b - a * b$ . Polje  $c$  mora zauzimati točno 4 bajta memorije.

**Napomena:** rješenje u kojem polje  $c$  zauzima više od 4 bajta memorije ili zadani elementi nisu točno spremljeni u memoriju donosi 0 bodova.

#### Rješenje:

```
#include<stdio.h>

int main(void){

int a = 5, b = 8;
signed char c[4];

c[0] = a+b;
c[1] = a-b;
c[2] = 2*a+b;
c[3] = a-2*b;

printf("%u\n",sizeof(c));

for(int i=0;i<4;i++)
    printf("%d ",c[i]);
printf("\n");

    return 0;
}
```

- (b) Objasnite značenje u programskom jeziku C:

1. `\x??` (upitnik reprezentira znamenku)
2. `\???` (upitnik reprezentira znamenku)
3. `scanf("%15[a-z]", str);`
4. `scanf("%i_%i_%i", &x, &y, &z);`
5. `i = (x = 10, x*=2);`
6. `*ptr++;`

- (c) Skicirajte i objasnite:

1. Prikaz broja  $-0$  u tipu `binary32`.
2. Prikaz broja  $-\infty$  u tipu `binary32`.

**Rješenja** potražiti u predavanjima kolegija programiranje 1.

# Programiranje 1 – ispit, 22. 2. 2024.

## Rješenja zadataka

**Zadatak 3** (25 bodova) Neka je  $n$  prirodan broj i  $p$  polinom definiran formulom

$$p(x) = \sum_{i=0}^{\lfloor n/2 \rfloor} \binom{n-i}{i} (3x)^{n-2i},$$

gdje s  $\lfloor x \rfloor$  označavamo najveći cijeli broj koji nije veći od  $x$ .

Napišite funkciju `izbaci` koja prima prirodne brojeve  $m$  i  $n$  te niz prirodnih brojeva  $a$  duljine  $m$ , te iz danog niza izbacuje sve elemente  $a_i$  takve da je  $p(a_i)$  prost. Funkcija vraća najmanji izbačeni element. Ako takav ne postoji, funkcija vraća  $-1$ . Dodatno, funkcija kroz varijabilni argument vraća novu duljinu niza.

**Napomena:** Možete definirati dodatne (pomoćne) funkcije. **Obavezno** je korištenje Hornerovog algoritma.

**Rješenje:**

```
int povrh(int n, int k){
    int i, prod=1;
    if(n<2*k){
        for(i=n ; i>k ; i--)
            prod *= i;
        for(i=1 ; i<=n-k ; i++)
            prod /= i;
    }
    else {
        for(i=n ; i>n-k ; i--)
            prod *= i;
        for(i=1 ; i<=k ; i++)
            prod /= i;
    }
    return prod;
}

double horner(double x, int n){
    double p=0;
    int k, i, z;
    if(n%2==0)
        k=n/2;
    else
        k=(n-1)/2;
    for(i=k ; i>=0 ; i--){
        if(n%2==0)
            z=povrh((n+2*i)/2, (n-2*i)/2);
        else
            z=povrh((n+2*i+1)/2, (n-2*i-1)/2);
        p=p*9*x*x+z;
    }
    if(n%2==1)
        p=p*3*x;
    return p;
}

int prost(int n){
    int i;
    for(i=2 ; i<n ; i++)
        if(n%i==0)
            return 0;
    return 1;
}

double izbaci(double a[], int n, int* m){
    double min=-1;
    int i;
    for(i=0 ; i<*m ; i++){
```

```
if(prost(a[i])){
    if(min==-1)
        min=a[i];
    else if(a[i]<min)
        min=a[i];
    int j;
    for(j=i; j<*m-1 ; j++)
        a[j]=a[j+1];
    (*m)--;
    i--;
}
}
return min;
}
```

## Programiranje 1 – ispit, 22. 2. 2024.

### Rješenja zadataka

**Zadatak 4** (4+4+4=12 bodova) Za uređenu trojku cijelih brojeva  $(x, y, z)$  definiramo njezin **pitagorejski defekt** kao vrijednost  $z^2 - y^2 - x^2$  te kažemo da je trojka Pitagorina ukoliko vrijedi  $z^2 = x^2 + y^2$ . Napišite funkciju **Pitagora** koja kao argumente prima tri niza cijelih brojeva **a**, **b** i **c** za koje pretpostavljamo da imaju jednako mnogo elemenata te nenegativni cijeli broj  $d \leq 100$  koji bilježi veličinu svakog od promatranih nizova.

Nizove interpretiramo tako da za dani  $i \in \{0, 1, 2, \dots, d-1\}$ ,  $(\mathbf{a}[i], \mathbf{b}[i], \mathbf{c}[i])$  predstavlja jednu trojku cijelih brojeva. Funkcija treba sortirati nizove **a**, **b** i **c** tako da sortirani nizovi i dalje predstavljaju iste trojke cijelih brojeva kao i nesortirani nizovi te tako da za  $i, j \in \{0, 1, 2, \dots, d-1\}$  takve da je  $i < j$  mora vrijediti da trojka  $(\mathbf{a}[i], \mathbf{b}[i], \mathbf{c}[i])$  ima pitagorejski defekt manji ili jednak od pitagorejskog defekta točke  $(\mathbf{a}[j], \mathbf{b}[j], \mathbf{c}[j])$ . Funkcija također treba vratiti broj pronađenih Pitagorinih trojki te kroz varijabilni argument treba vratiti najmanju vrijednost pitagorejskog defekta. Napišite i program koji će učitati tri niza cijelih brojeva te pozvati funkciju **Pitagora**.

**Rješenje:**

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

int Pitagora(int x[], int y[], int z[], int d, int *def )
{
    int i,j,tx,ty,tz,Pit,min;
    int defi,defj;
    Pit=0;
    min=0;
    for(i=0; i<d-1; i++)
    {
        for(j=i+1;j<d; j++)
        {
            defi=z[i]*z[i]-y[i]*y[i]-x[i]*x[i];
            defj=z[j]*z[j]-y[j]*y[j]-x[j]*x[j];
            if(defi>defj)
            {
                tx=x[j];
                ty=y[j];
                tz=z[j];
                x[j]=x[i];
                y[j]=y[i];
                x[i]=tx;
                y[i]=ty;
                z[i]=tz;
            }
        }
    }
    for(int i=0;i<d; i++)
    {
        defi=z[i]*z[i]-y[i]*y[i]-x[i]*x[i];
        if(defi<min) min=defi;
        if(defi==0) Pit++;
    }

    *def= min;
    return Pit;
}

int main()
{
    int rez1, rez2;
    int x[100];
```

```
int y[100];
int z[100];
printf("Unesite broj tocaka u ravnini koje sortiramo \n");
int d;
scanf("%d", &d);
for(int i = 0; i<d ;i++)
{
    printf("Unesite x koordinatu tocke %d \n", i);
    scanf("%d", &x[i]);
    printf("Unesite y koordinatu tocke %d \n", i);
    scanf("%d", &y[i]);
    printf("Unesite z koordinatu tocke %d \n", i);
    scanf("%d", &z[i]);
}

rez1=Pitagora(x,y,z,d,&rez2);
printf("%d %d", rez1, rez2);
return 0;
}
```

# Programiranje 1 – ispit, 22. 2. 2024.

## Rješenja zadataka

**Zadatak 5** (5+15=20 bodova) Za niz cijelih brojeva  $S = \{a[0], a[1], \dots, a[n-1]\}$  kažemo da je *konveksan* ako vrijedi

$$a[i+2] - 2a[i+1] + a[i] \geq 0$$

za svaki  $i \in \{0, 1, \dots, n-3\}$ . Podrazumijevamo da je svaki niz duljine 1 ili 2 konveksan.

- (a) Napišite funkciju `int is_convex(int a[], int n)` koja prima niz cijelih brojeva  $(a[0], a[1], \dots, a[n-1])$  i njegovu duljinu  $n$  te provjerava je li niz konveksan. Ako jest, funkcija vraća 1, a inače funkcija vraća 0.
- (b) Napišite funkciju `int longest_convex_subseq(int a[], int n)` koja prima niz cijelih brojeva  $(a[0], a[1], \dots, a[n-1])$  i njegovu duljinu  $n$ , te vraća duljinu najvećeg konveksnog podniza, odnosno najveći prirodan broj  $m$  takav da postoji  $m$  uzastopnih članova danog niza koji čine konveksan niz. Smijete koristiti funkciju iz (a) dijela zadatka čak i ako ju niste napisali.
- (c) (Dodatnih 10 bodova koji se ne ubrajaju u uvjet od 80 posto) Napišite funkciju `void sort_convex(int a[], int n)` koja prima konveksan niz cijelih brojeva  $(a[0], a[1], \dots, a[n-1])$  i njegovu duljinu  $n$ , te ga sortira uzlazno. Smijete koristiti pomoćna polja tipa `int`. Zadatak treba riješiti u **linearnoj složenosti**.

### Rješenje:

```
(a) int is_convex(int a[], int n){
    int i;
    for (i=0; i<n-2;i++) if (a[i+2]-2a[i+1]+a[i]<0) return 0;
    return 1;
}

(b) int longest_convex_subseq(int a[], int n){
    int maxlength=0;
    int i,j;
    for (i=0;i<n;i++){
        for (j=i;j<n;j++){
            if (is_convex(a+i, j-i+1) && j-i+1>maxlength) maxlength=j-i+1;
        }
    }
    return maxlength;
}

(c) void sort_convex(int a[], int n){
    int min=a[0], argmin=0, i, j;
    for (i=0;i<n;i++){
        if (a[i]<min){
            min=a[i];
            argmin=i;
        }
    }
    int b[1000];
    int c[1000];
    int lb=argmin, lc=n-argmin;
    for (i=0;i<lb;i++) b[i]=a[argmin-i-1];
    for (i=0;i<lc;i++) c[i]=a[i+argmin];
    i=0;
    j=0;
    while (i<lb && j<lc){
        if (b[i]<c[j]){
            a[i+j]=b[i];
            i++;
        }
        else{
            a[i+j]=c[j];
            j++;
        }
    }
}
```

```
    }  
}  
  
while (i<lb){  
    a[i+j]=b[i];  
    i++;  
}  
  
while (j<lc){  
    a[i+j]=c[j];  
    j++;  
}  
return;  
}
```