

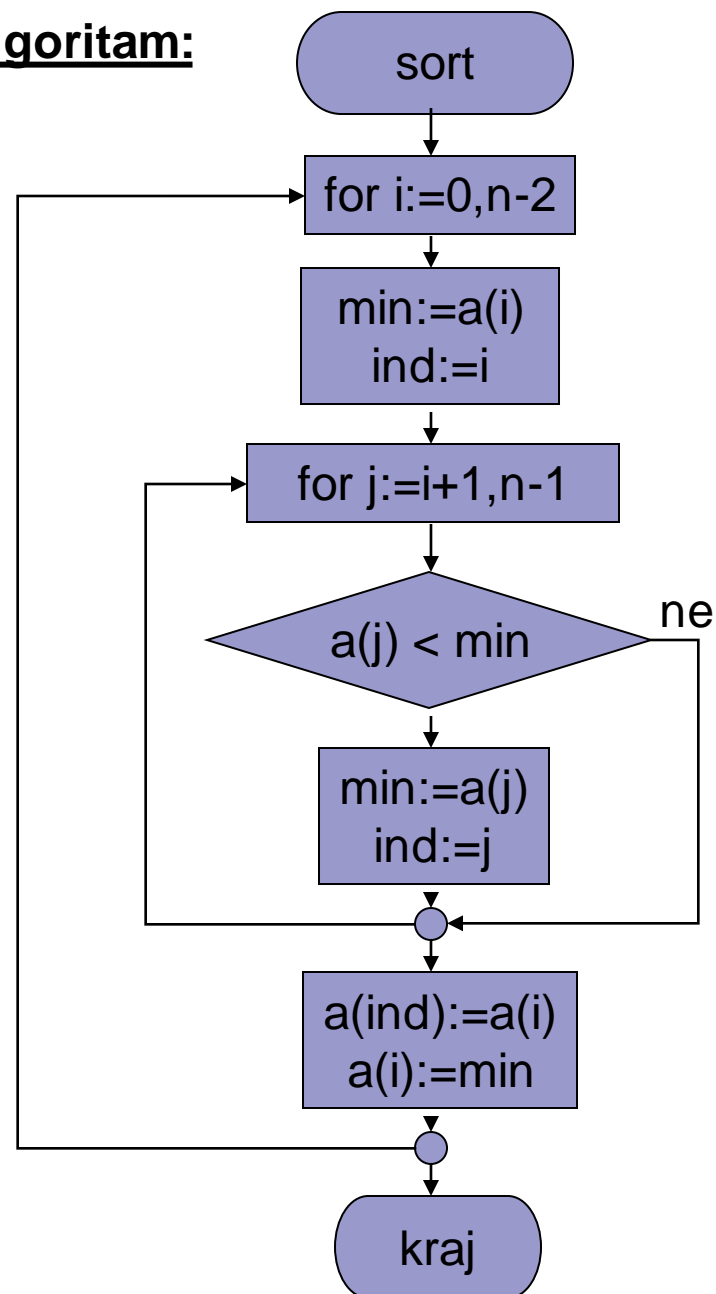


# Programski jezik C

## Primeri

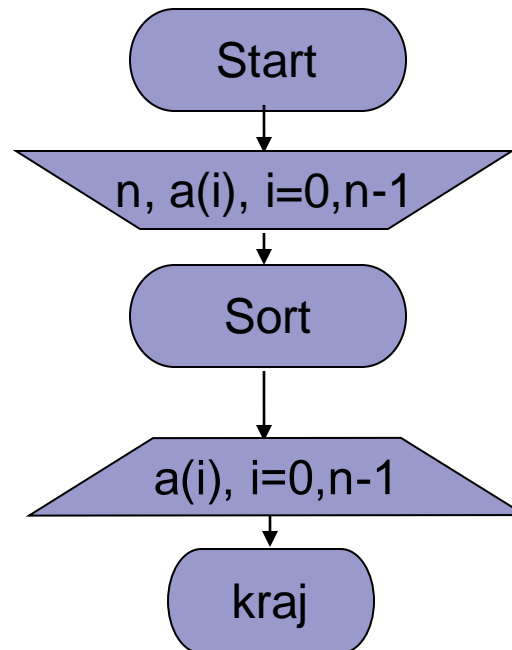
**Zadatak:**

Napisati algoritam i funkciju na C-u koja vrši sortiranje celobrojnog niza **a** u neopadajući redosled, kao i glavni program koji poziva funkciju.

**Algoritam:**

**C funkcija:**

```
#include <stdio.h>
#define MAXSIZE 100
int a[MAXSIZE], n;
void sort(void)
{
    int i, imin, min, j;
    for (i = 0; i < n-2; i++) {
        ind = i; min = a[i];
        for (j = i+1; j <= n-1; j++)
            if (a[j]<min) {
                ind = j;
                min = a[j];
            }
        a[ind] = a[i];
        a[i] = min;
    }
}
```

**Glavni program:**

**Glavni program (C):**

```
void main(void)
{
    int i;
    printf("Unesi velicinu niza: ");
    scanf("%d", &n);
    printf("Unesi %d celih brojeva:\n", n);
    for (i = 0; i < n; i++)
        scanf("%d", &a[i]);
    sort();
    printf("Sortirani niz:\n");
    for (i=0; i<n; i++)
        printf("%d\n", a[i]);
}
```

**Zadatak:**

Napisati algoritam i funkciju na C-u koja vrši sortiranje celobrojnog niza **a** u neopadajući redosled korišćenjem pointera, kao i glavni program koji poziva funkciju.

**Rešenje:**

```
#include <stdio.h>
#define MAXSIZE 100
int a[MAXSIZE], size;
void sort(void)
{
    int *p, *pmin, min, *q, *pn = a + size-1;
    for (p = a; p < pn-1; p++) {
        pmin = p; min = *p;
        for (q = p+1; q <= pn; q++)
            if (*q < min) {
                pmin = q;
                min = *q;
            }
        if (pmin != p) {
            *pmin = *p;
            *p = min;
        }
    }
}
```

**Glavni program (C):**

```
void main(void)
{
    int *p;
    printf("Unesi velicinu niza: ");
    scanf("%d", &size);
    printf("Unesi %d celih brojeva:\n", size);
    for (p = a; p < a+size; p++)
        scanf("%d", p);
    sort();
    printf("Sortirani niz:\n");
    for (p = a; p < a+size; p++)
        printf("%d\n", *p);
}
```

**Zadatak:**

Napisati funkciju na C-u koja proverava da li je broj prost. U glavnom programu naći sumu prostih brojeva koji su manji od zadanog broja.

**Rešenje:**

```
/* Vрати 1 ako je prost broj, 0 ako nije. */

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

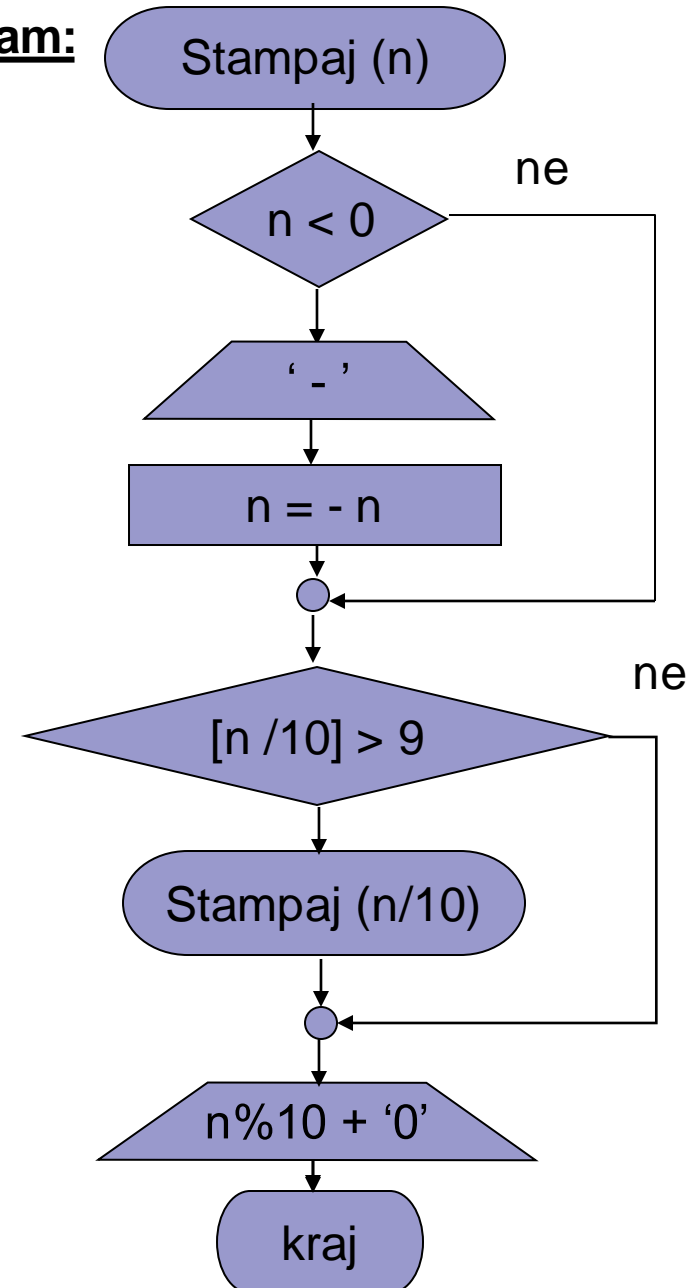


**Glavni program:**

```
#include <stdio.h>
int isPrime (int n);
int main () {
    int n, limit , sum = 0;
    printf (" Unesi granicu : ");
    scanf ("%d", &limit );
    for (n = 1; n <= limit ; ++n) {
        if ( isPrime (n)) {
            sum += n;
        }
    }
    printf ("Suma prostih brojeva <= %d je %d\n",limit , sum );
    return 0;
}
```

**Zadatak:**

Napisati algoritam i funkciju na C-u koja vrši štampanje cifara zadatog celog broja.

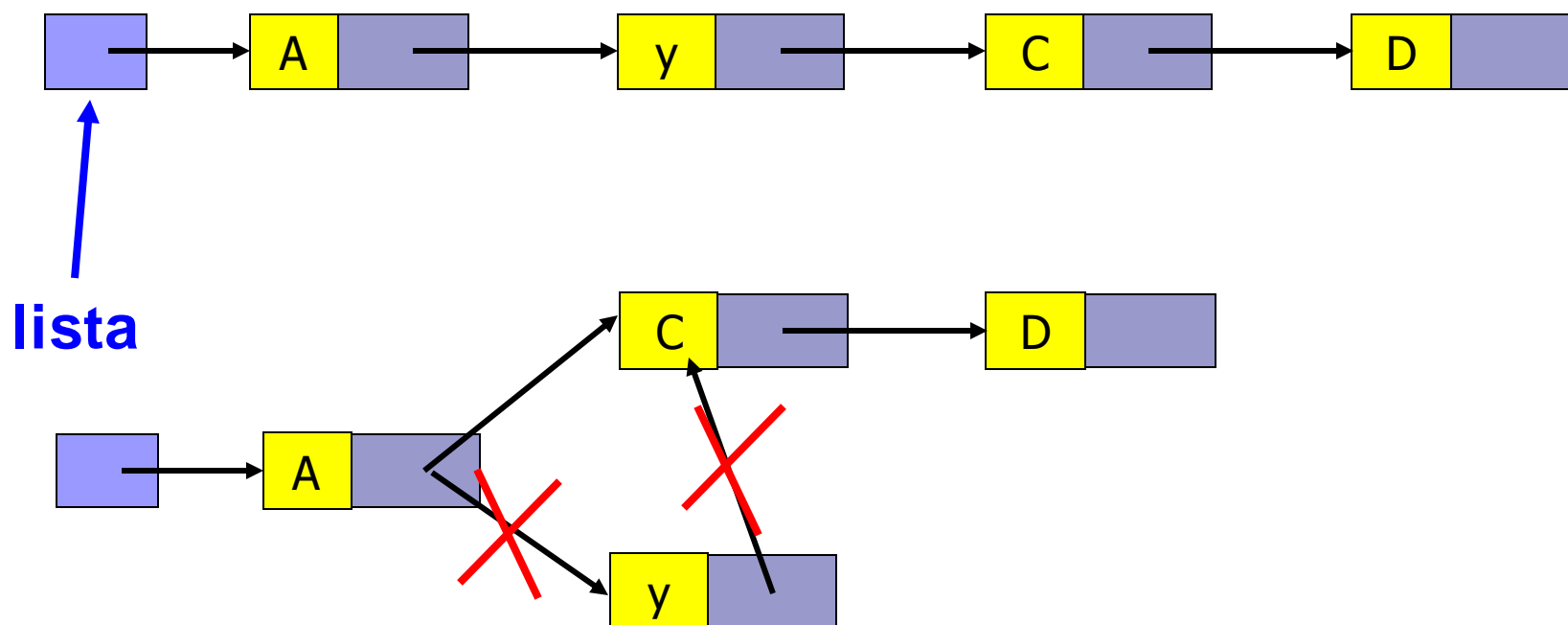
**Algoritam:**

C funkcija:

```
#include <stdio.h>
void stampaj(int n)
{
    if (n < 0) {
        printf("%c", '-');
        n = -n;
    }
    if (n/10)
        stampaj(n/10);
    printf("%c", n%10 + '0');
}
```

**Zadatak: Lančane liste**

Napisati funkciju na C-u koja briše prvi čvor sa vrednošću **y** iz lančane liste **lista**.

**Rešenje:**

**Rešenje:**

```
#include <stdio.h>
#include <stdlib.h>
#include <malloc.h>
struct cvor{
    int sadrzaj;
    cvor *sledeći;
};
void brisi(struct cvor *lista, int y){
    struct cvor *p, *prethodni;
    if (lista->sadrzaj == y) lista = lista->sledeci;
    else {
        prethodni = lista; p = prethodni->sledeci;
        while (p->sadrzaj != y && p != NULL){
            prethodni = p;
            p = p->sledeci;
        }
        if (p != NULL){
            prethodni->sledeci = p->sledeci;
            free (p);
        }
        else
            printf("Ne postoji elemenat sa zadatim kljucem\n");
    }
}
```

**Zadatak:**

Napisati funkciju na C-u koja vrši obrtanje (inverziju) stringa.

**Rešenje:**

```
#include <stdio.h>
void obrni(char s[])
{
    int c,i,j;
    for(i = 0, j = strlen(s)-1; i < j; i++, j--){
        c = s[i];
        s[i] = s[j];
        s[j] = c;
    }
}
```

**Zadatak:**

Napisati funkciju na C-u koja vrši kopiranje sadržaja jedne datoteke u drugu.

**Rešenje:**

```
#include <stdio.h>
main()
{
    char in_name[25], out_name[25];
    FILE *in_file, *out_file;
    int c;

    printf("Datoteka koja se kopira:\n");
    scanf("%24s", in_name);
    printf("Ime datoteke u koju se kopira:\n");
    scanf("%24s", out_name);
    in_file = fopen ( in_name, "r");
    if( in_file == NULL )
        printf("Ne može se otvoriti %s za čitanje.\n", in_name);
    else {
        out_file = fopen (out_name, "w");
        if( out_file == NULL )
            printf("Ne može se otvoriti %s za pisanje.\n",out_name);
        else {
            while( (c = getc( in_file)) != EOF )
                putc (c, out_file);
            putc (c, out_file); /* copy EOF */
            printf("Datoteka je kopirana.\n");
            fclose (out_file);
        }
        fclose (in_file);
    }
}
```

**Zadatak:**

Napisati funkciju na C-u koja čita redom linije iz teksta čiji je kraj označen praznom linijom, sortira ih i štampa.

**Rešenje:**

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#define MAXSIZE 100
#define MAXLEN 80
char array[MAXSIZE][MAXLEN];
int size;
void sort(void)
{
    int i, imin, j, n = size-1;
    char min[MAXLEN];
    for (i = 0; i < n; i++) {
        imin = i; strcpy(min, array[i]);
        for (j = i+1; j <= n; j++)
            if (strcmp(array[j], min) < 0) {
                imin = j;
                strcpy(min, array[j]);
            }
        if (imin != i) {
            strcpy(array[imin], array[i]);
            strcpy(array[i], min);
        }
    }
}
```

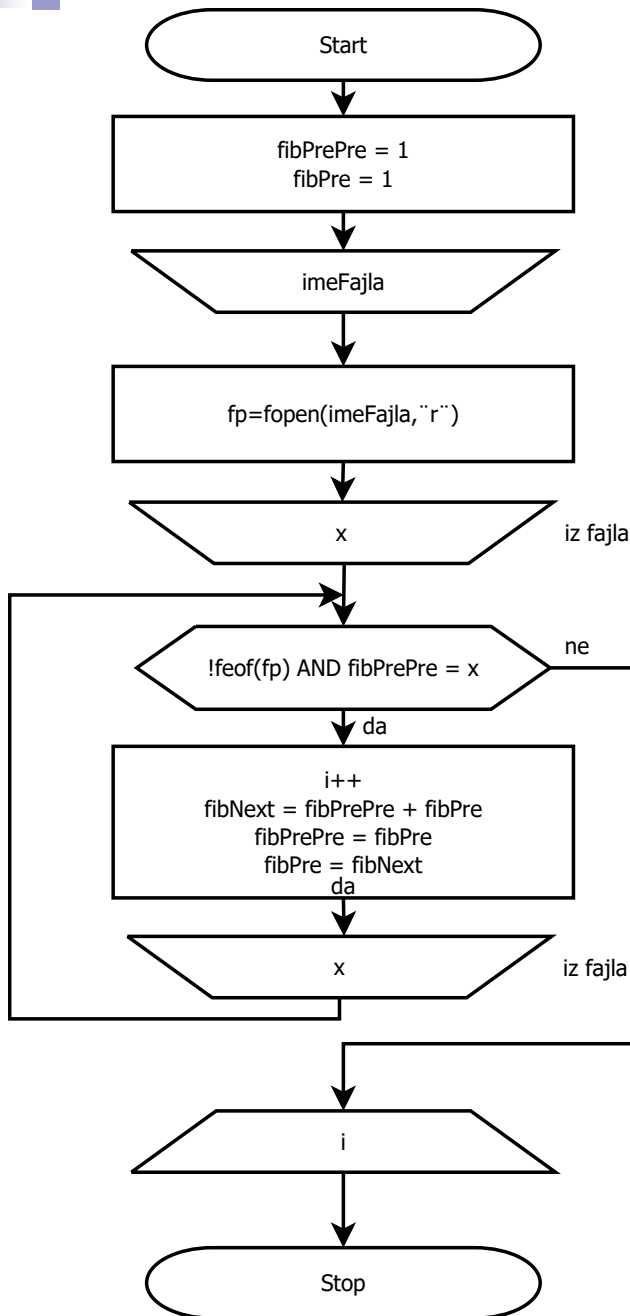


**Glavni program:**

```
void main(void)
{
    int i;
    size = 0;
    printf("Unesi tekst završen praznom linijom:\n");
    gets(array[size]);
    while (strlen(array[size]))
        gets(array[++size]);
    sort();
    printf("Sortirane linije teksta:\n");
    for (i = 0; i < size; i++)
        printf("%s\n", array[i]);
}
```

# Zad 1:

- U tekstualnom fajlu smešteni su celi brojevi. Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati strukturni program koji određuje koliko brojeva od početka fajla redom se poklapa sa Fibonačijevim nizom. Elemente Fibonačijevog niza, u cilju poređenja sa elementima iz fajla, izračunavati na sledeći način:  $f_i = f_{i-1} + f_{i-2}$ , za  $i \geq 3$ , pri čemu su  $f_1 = 1$  i  $f_2 = 1$ . Napomene: Nije poznat broj brojeva u fajlu. Nije dozvoljeno korišćenje indeksiranih promenljivih.



```

#include <stdio.h>
void main()
{
    int i=0, fibPrePre=1, fibPre=1, fibNext, x;
    char imeFajla[100]; //ili, ako je fiksirano: = "fajl.txt";
    FILE *fp;

    printf("Unesite ime fajla:\n");
    gets(imeFajla);
    fp = fopen(imeFajla,"r"); //ili, ako je fiksirano: fopen("fajl.txt","r");

    fscanf(fp, "%d", &x);
    while (!feof(fp) && fibPrePre == x)
    {
        i++;
        fibNext = fibPrePre + fibPre;
        fibPrePre = fibPre;
        fibPre = fibNext;
        fscanf(fp, "%d", &x);
    }
    printf("Prvih %d brojeva iz fajla se poklapa sa Fibonaccievim nizom.\n", i);
}

```

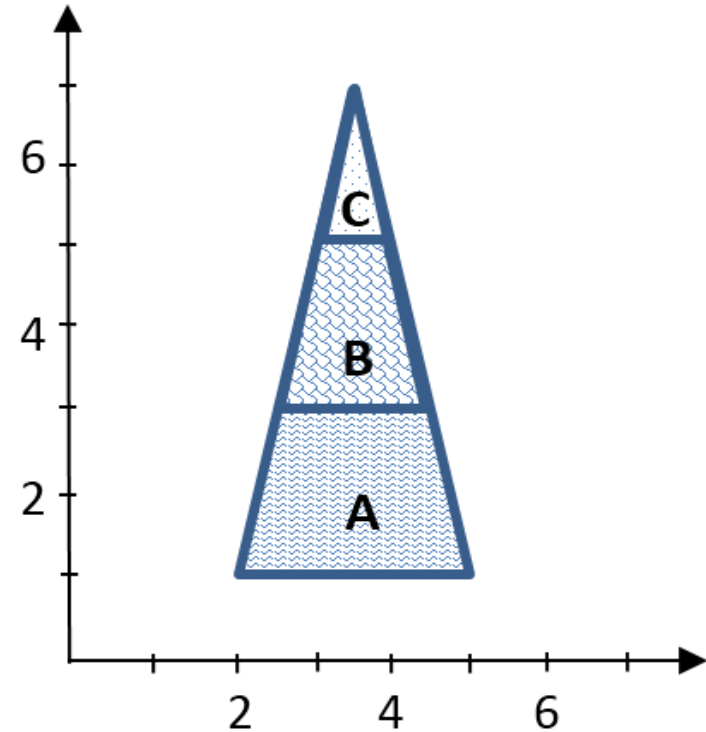
# Zad:

- N timova planinara započelo je penjanje na visoku planinu sačinjenu od tri oblasti: zelene, krševite i snežne oblasti. Nacrtati strukturni dijagram toka algoritma i napisati strukturni program na programskom jeziku C kojim se određuje koliko timova je stiglo do svake od oblasti na planini. Pozicije timova zadate su svojim koordinatama  $(x, y)$ . Oblasti planine prikazane su na slici: zelena

```
#include <stdio.h>
void main() {
    int i,N,obA,obB,obC;
    float x,y;
    i=0;
    obA=0;

obB=0;
obC=0;
scanf("%d", &N);
    for(i=0;i<N;i++)
    {
        scanf("%f %f", &x,&y);

        if((y<=4*x-7.0)&&(y<=-4*x+21.0))
            if(y>1&&y<3)
                obA++;
            else if(y>=3&&y<5)
                obB++;
            else if(y>=5)
                obC++;
    }
    printf("Broj timova koji je stigao do zelene oblasti je: %d",obA);
    printf("Broj timova koji je stigao do krsevite oblasti je: %d",obB);
    printf("Broj timova koji je stigao do snezne oblasti je: %d",obC);
}
```



# ZAD:

Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati strukturnu funkciju koja izračunava faktoriyel prirodnog broja prosleđenog preko parametara . U glavnom programu uneti broj elemenata  $N$  niza celih brojeva  $A$ . Elemente niza zatim generisati tako da je vrednost svakog elementa jednaka faktoriyelu njegovog indeksa. Za izračunavanje faktoriyela koristiti napisanu funkciju. Prikazati svaki element niza u formatu:  $A[<indeks\_elementa>] = <vrednost\_elementa>$ .

Nacrtati strukturni dijagram toka algoritma i na programskom jeziku C napisati strukturni program kojim se određuje suma elemenata u svakoj od vrsta celobrojne kvadratne matrice  $\mathbf{A}_{N \times N}$ . Nakon toga, potrebno je urediti vrste matrice  $\mathbf{A}$  u nerastući redosled na osnovu suma njihovih elemenata. Red  $N$ , kao i vrednosti elemenata matrice  $\mathbf{A}$  zadaje korisnik. Prikazati matricu  $\mathbf{A}$  pre i posle uređenja.

Primer:

$$\text{Ulaz: } N = 5, \mathbf{A} = \begin{bmatrix} 1 & 1 & 2 & 3 & 5 \\ -2 & 3 & -5 & 7 & 11 \\ 8 & 3 & 6 & 5 & 1 \\ 10 & -8 & -6 & -4 & 2 \\ 1 & 5 & 15 & 3 & 7 \end{bmatrix} \begin{array}{l} \Sigma = 12 \\ \Sigma = 14 \\ \Sigma = 23 \\ \Sigma = -6 \\ \Sigma = 31 \end{array} \quad \text{Izlaz:}$$

$$\mathbf{A} = \begin{bmatrix} 1 & 5 & 15 & 3 & 7 \\ 8 & 3 & 6 & 5 & 1 \\ -2 & 3 & -5 & 7 & 11 \\ 1 & 1 & 2 & 3 & 5 \\ 10 & -8 & -6 & -4 & 2 \end{bmatrix} \begin{array}{l} \Sigma = 31 \\ \Sigma = 23 \\ \Sigma = 14 \\ \Sigma = 12 \\ \Sigma = -6 \end{array}$$