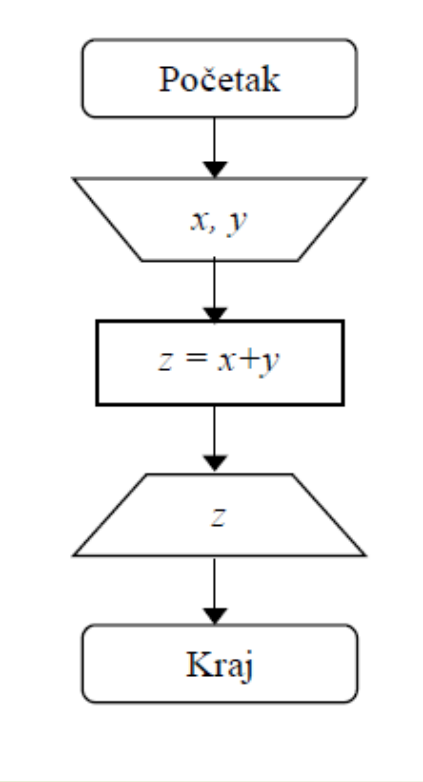
ALGORITMI I STRUKTURE PODATAKA

RAČUNSKE VEŽBE – TERMIN BR. 1 – 1

ALDINA AVDIĆ, DIPL. INŽ.

Prosta linijska struktura

- **Primer 1.** Nacrtati algoritam za sabiranje brojeva x i y .

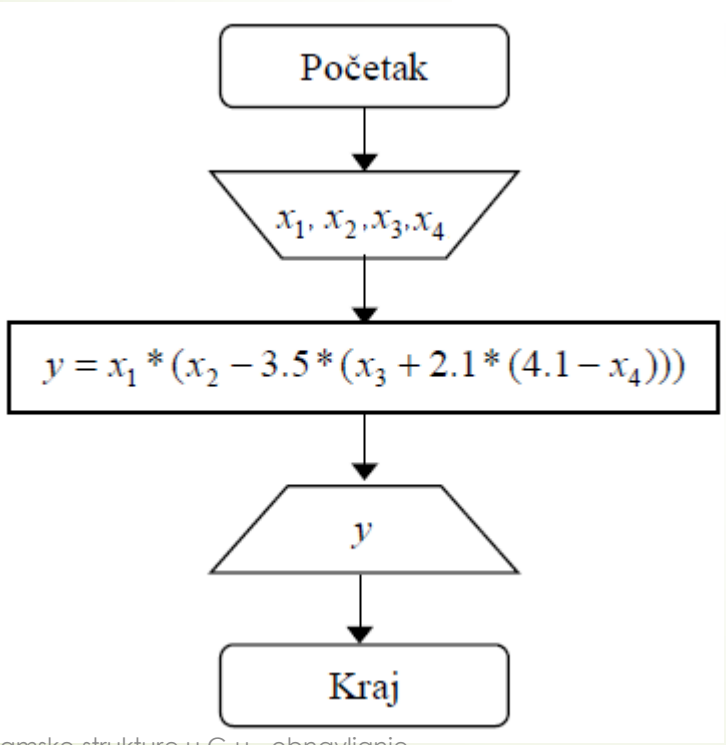



```c
#include <stdio.h>
int main()
{
    int x, y, z;
    printf("Unesite brojeve x i y \n");
    scanf("%d %d", &x, &y);
    z=x+y;
    printf("Rezultat je %d",z);
    system("PAUSE");
    return 0;
}
```

Prosta linijska struktura

- **Primer 2.** Nacrtati algoritam za iračunavanje sledeće funkcije:

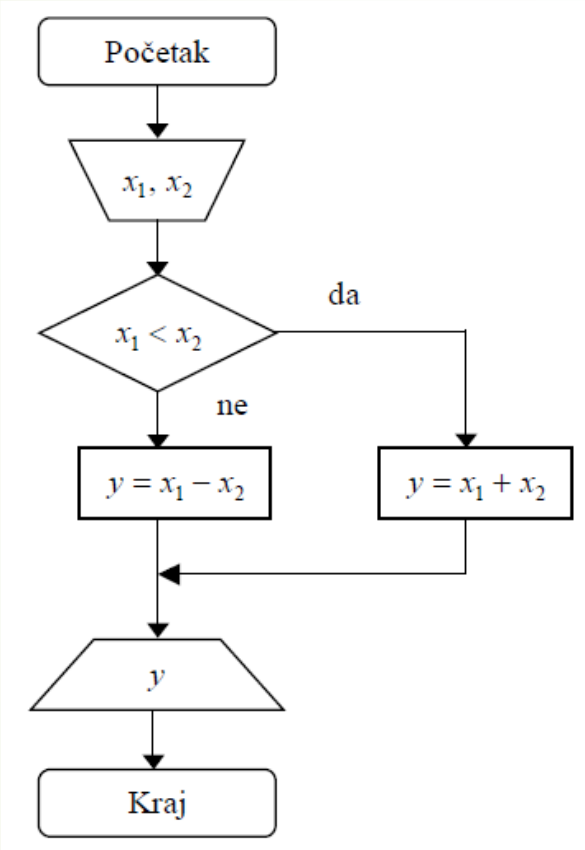
$$y = x_1\{x_2 - 3.5[x_3 + 2.1(4.1 - x_4)]\}$$




```c
#include <stdio.h>
int main()
{
    float x1, x2, x3, x4, y;
    printf("Unesite brojeve x1, x2, x3, x4 \n");
    scanf("%f %f %f %f", &x1, &x2, &x3, &x4);
    y=x1*(x2-3.5*(x3+2.1*(4.1-x4)));
    printf("Rezultat je %f",y);
    system("PAUSE");
    return 0;
}
```

Naredba grananja

- **Primer 3.** Nacrtati algoritam za iračunavanje sledeće funkcije:

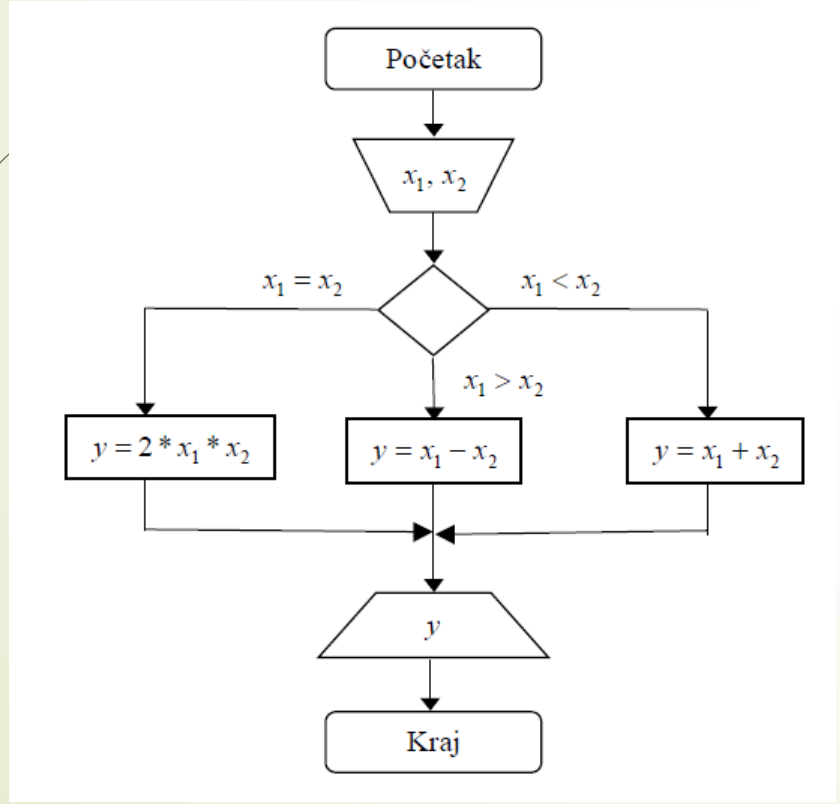


$$y = \begin{cases} x_1 + x_2 & ako\ je \quad x_1 < x_2 \\ x_1 - x_2 & ako\ je \quad x_1 \geq x_2 \end{cases}$$

```c
#include <stdio.h>
int main()
{
    int x1, x2, y;
    printf("Unesite brojeve x1 i x2 \n");
    scanf("%d %d", &x1, &x2);
    if(x1>x2)
    {
    y=x1+x2;
    }
    else
    {
        y=x1-x2;
    }
    printf("Rezultat je %d \n",y);
    system("PAUSE");
    return 0;
}
```

Naredba grananja

- **Primer 4.** Nacrtati algoritam za iračunavanje sledeće funkcije:

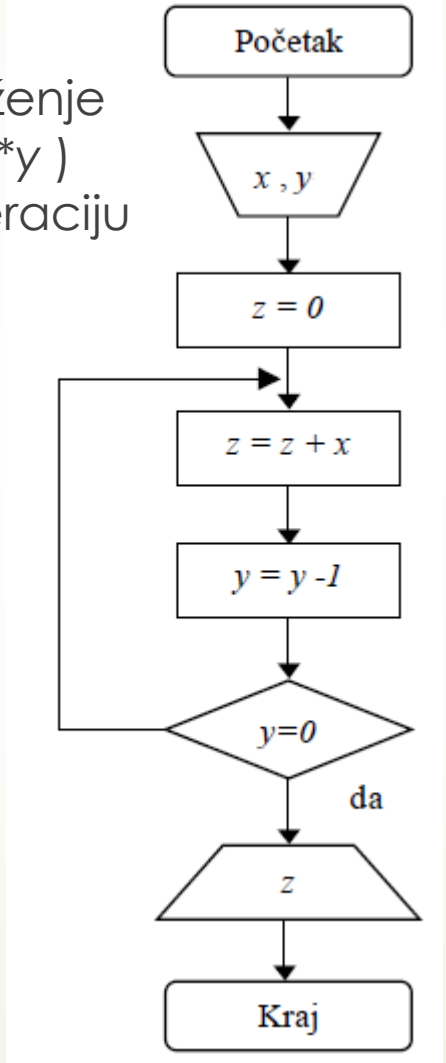


$$y = \begin{cases} x_1 + x_2 & ako\ je \quad x_1 < x_2 \\ x_1 - x_2 & ako\ je \quad x_1 > x_2 \\ 2x_1x_2 & ako\ je \quad x_1 = x_2 \end{cases}$$

```c
#include <stdio.h>
int main()
{
    int x1, x2, y;
    printf("Unesite brojeve x1 i x2 \n");
    scanf("%d %d", &x1, &x2);
    if(x1<x2)
    {
    y=x1+x2;
    }
    else if(x1>x2)
    {
        y=x1-x2;
    }
    else
    {
    y=2*x1;
    }
    printf("Rezultat je %d \n",y);
    system("PAUSE");
    return 0;
}
```

Do-While petlja

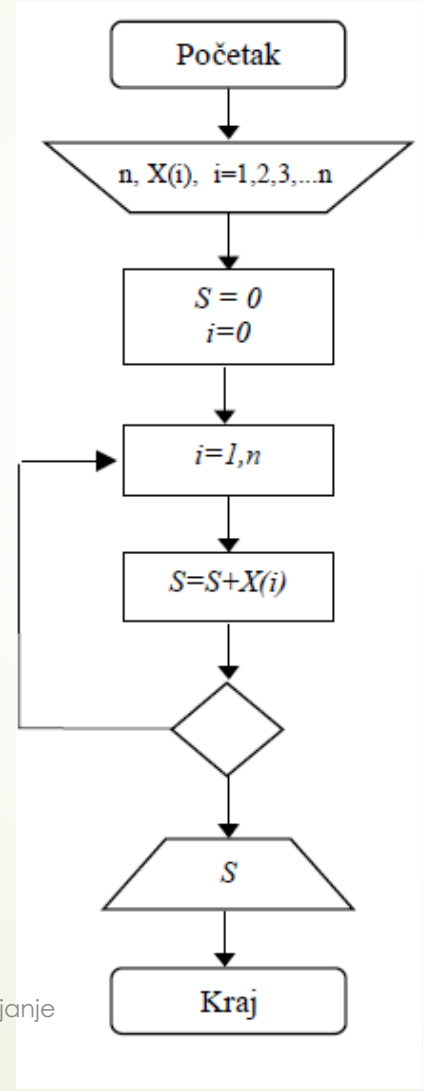
- **Primer 5.** Nacrtati algoritam za množenje dva prirodna (z=x*y) broja koristeći operaciju sabiranja.




```c
#include <stdio.h>
int main()
{
    int x, y, z;
    printf("Unesite brojeve x i y \n");
    scanf("%d %d", &x, &y);
    z=0;
    do
    {
    z=z+x;
    y=y-1;
    }
    while(y!=0);
    printf("Rezultat je %d",z);
    system("PAUSE");
    return 0;
}
```

For petlja

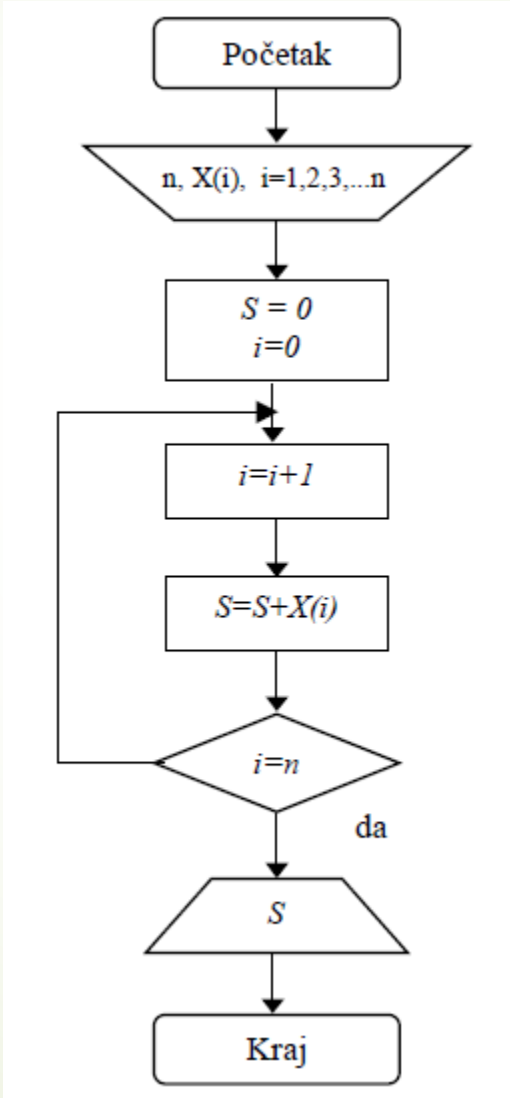
- **Primer 6.** Nacrtati algoritam za sabiranje svih članova niza *X(i), i=1,2,3,...n* .




```c
#include <stdio.h>
int main()
{
    int X[100], n, S, i;
    printf("Unesite n \n");
    scanf("%d", &n);
    printf("Unesite clanove niza \n");
    for(i=0;i<n;i++)
    {
    scanf("%d", &X[i]);
    }
    S=0;
    for(i=0;i<n;i++)
    {
    S=S+X[i];
    }
    printf("Rezultat je %d",S);
    system("PAUSE");
    return 0;
}
```

While petlja

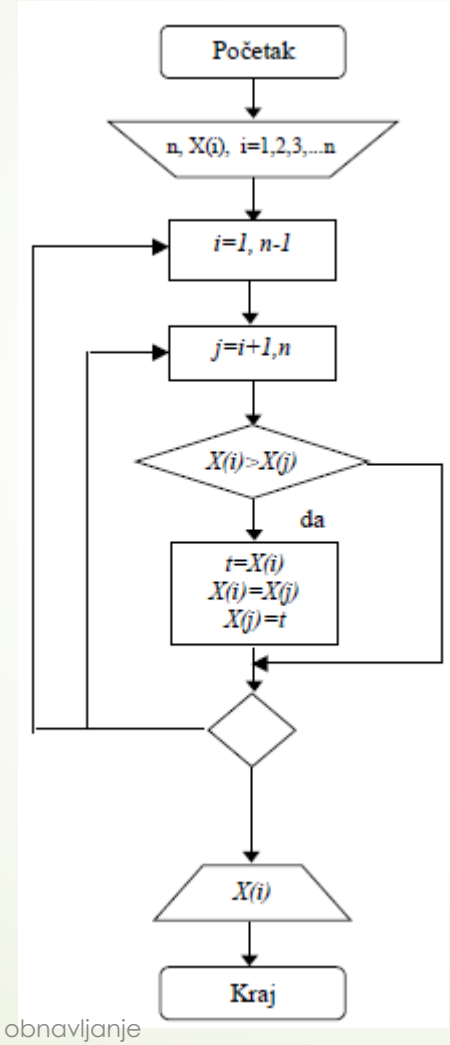
- **Primer 7.** Nacrtati algoritam za sabiranje svih članova niza *X(i), i=1,2,3,...n* .




```c
#include <stdio.h>
int main()
{
    int X[100], n, S, i;
    printf("Unesite n \n");
    scanf("%d", &n);
    printf("Unesite clanove niza \n");
    for(i=0;i<n;i++)
    {
    scanf("%d", &X[i]);
    }
    S=0;i=0;
    while(i<n)
    {
    S=S+X[i];
    i++;
    }
    printf("Rezultat je %d",S);
    system("PAUSE");
    return 0;
}
```

Sortiranje niza

- **Primer 8.** Nacrtati algoritam koji dati niz *X(i), i=1,2,3,...n* sortira u rastući.

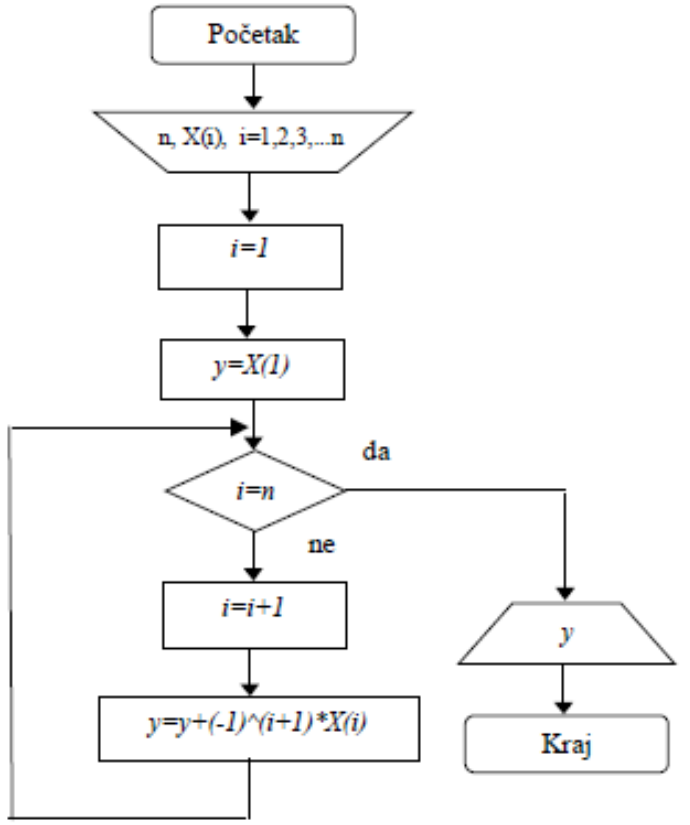



```c
#include <stdio.h>
int main()
{
    int X[100], n, pom, i, j;
    printf("Unesite n \n");
    scanf("%d", &n);
    printf("Unesite clanove niza \n");
    for(i=0;i<n;i++)
    {
    scanf("%d", &X[i]);
    }
    for(i=0;i<n;i++)
    {
        for(j=i+1;j<n;j++)
    {
    if(X[i]>X[j])
    {
        pom=X[i];
        X[i]=X[j];
        X[j]=pom;
    }
    }
    }
    printf("Sortirani niz");
    for(i=0;i<n;i++)
    {printf("%d", X[i]);
    }
    system("PAUSE");
    return 0;
}
```

Niz

- **Primer 9.** Nacrtati algoritam za određivanje sledeće sume:

$$y = \sum_{i=1}^{n} (-1)^{i+1} X(i); \qquad n \geq 1$$

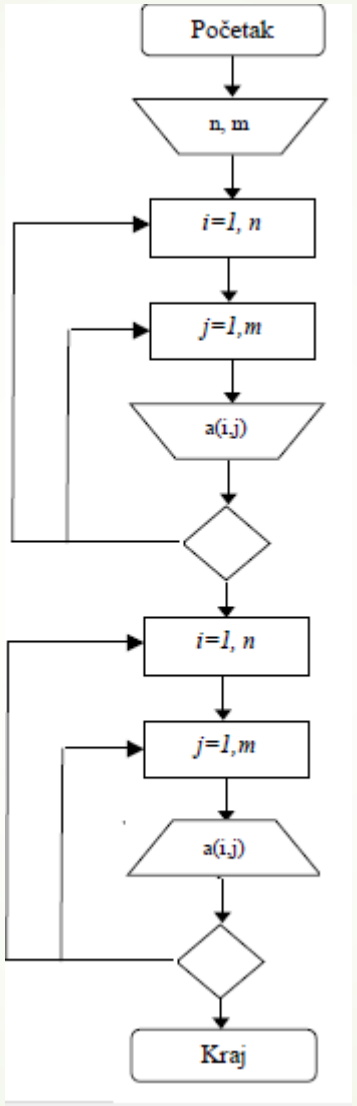



```
#include <stdio.h>
int main()
{
    int X[100], n, y, i, j;
    printf("Unesite n \n");
    scanf("%d", &n);
    printf("Unesite clanove niza \n");
    for(i=0;i<n;i++)
    {
    scanf("%d", &X[i]);
    }
    y=0;
    for(i=0;i<n;i++)
    {
    y+=pow(-1,i)*X[i];
    }

    printf("Rezultat je %d", y);

    system("PAUSE");
    return 0;
}
```

Matrica

- **Primer 10.** Nacrtati algoritam za učitavanje i ispis svih elemenata matice A reda *m* x *n*.

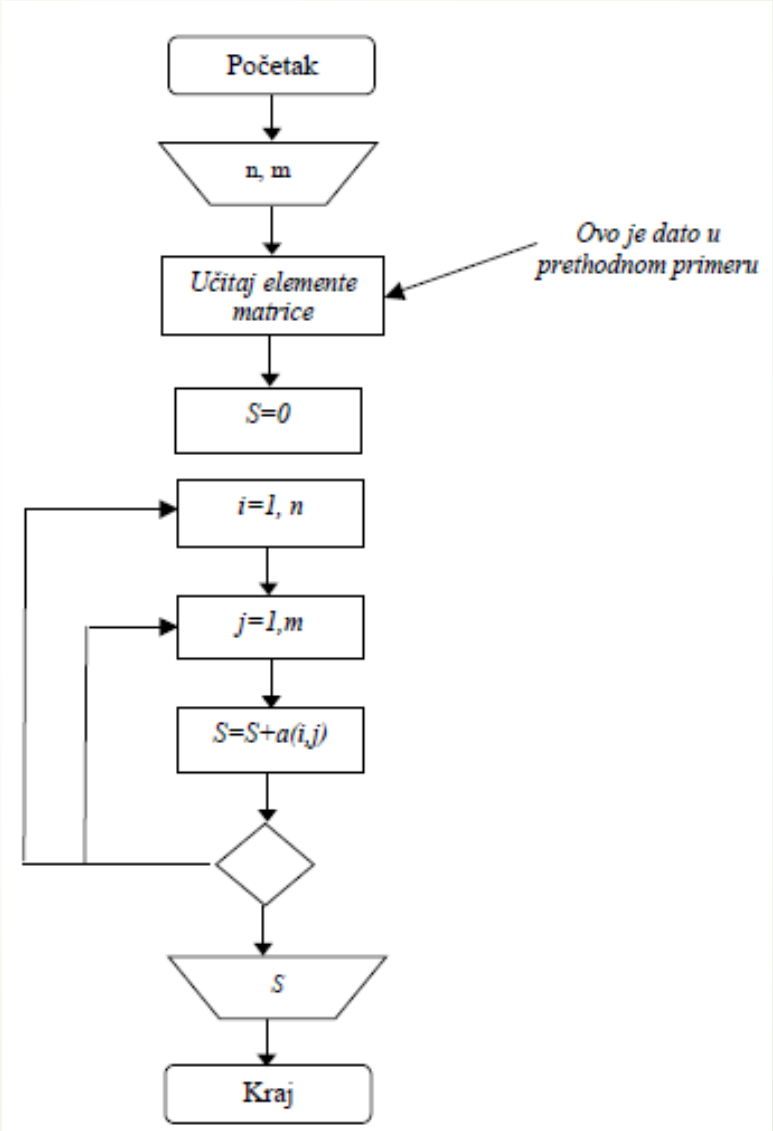
$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \dots & a_{3n} \\ \dots & \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & a_{m3} & \dots & a_{mn} \end{pmatrix}$$




```c
#include <stdio.h>
int main()
{
    int A[10][10], n,m, i, j;
    printf("Unesite n \n");
    scanf("%d", &n);
        printf("Unesite m \n");
    scanf("%d", &m);
    printf("Unesite elemente matrice \n");
    for(i=0;i<n;i++)
    {
        for(j=0;j<m;j++)
    {
    scanf("%d", &A[i][j]);
    }}
    for(i=0;i<n;i++)
    {
        for(j=0;j<m;j++)
    {
    printf("%d ", A[i][j]);
    }
    printf("\n");
    }
    system("PAUSE");
    return 0;
}
```

Matrica

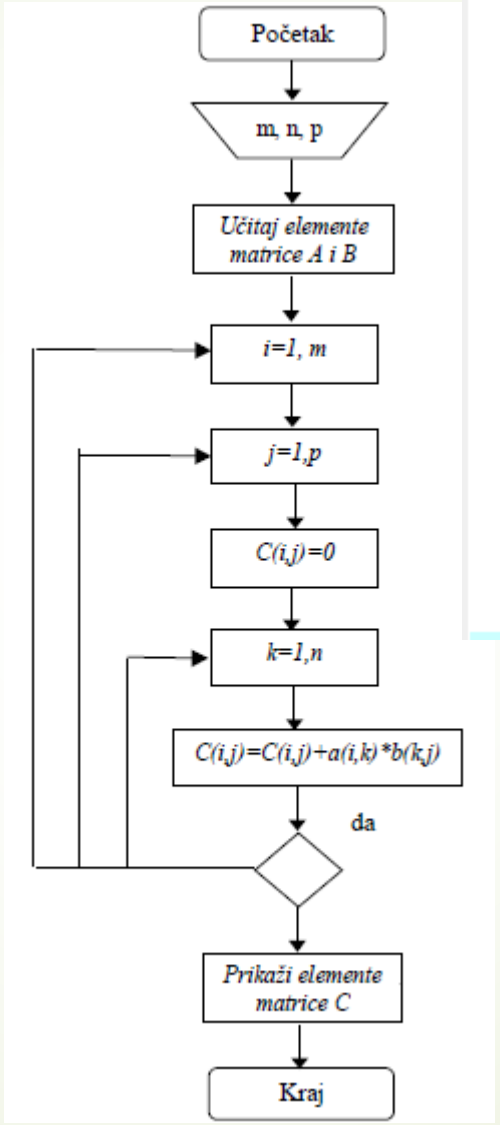
- **Primer 11.** Nacrtati algoritam za sabiranje svih elemenata matice A reda *m* x *n*.




```c
#include <stdio.h>
int main()
{
    int A[10][10], n,m, i, j, S=0;
    printf("Unesite n \n");
    scanf("%d", &n);
        printf("Unesite m \n");
    scanf("%d", &m);
    printf("Unesite elemente matrice \n");
    for(i=0;i<n;i++)
    {
        for(j=0;j<m;j++)
    {
    scanf("%d", &A[i][j]);
    }}
    for(i=0;i<n;i++)
    {
        for(j=0;j<m;j++)
    {
    S+=A[i][j];
    }
    }
    printf("S=%d", S);
    system("PAUSE");
    return 0;
}
```

Matrica

- **Primer 12.** Nacrtati algoritam za množenje matice A reda (*m* x *n*) i matrice B (reda *n* x *p*).




```c
#include <stdio.h>
int main()
{
    int A[10][10],B[10][10],C[10][10], n,m,p, i, j,k, S=0;
    printf("Unesite n \n");
    scanf("%d", &n);
        printf("Unesite m \n");
    scanf("%d", &m);
    printf("Unesite p \n");
    scanf("%d", &p);
    printf("Unesite elemente matrice A \n");
    for(i=0;i<n;i++)
    {
        for(j=0;j<m;j++)
    {
    scanf("%d", &A[i][j]);
    }}
    printf("Unesite elemente matrice B \n");
    for(i=0;i<m;i++)
    {
        for(j=0;j<p;j++)
    {
    scanf("%d", &B[i][j]);
    }}
```

```c
for(i=0;i<n;i++)
{
    for(j=0;j<p;j++)
{
    C[i][j]=0;
    for (k=0;k<m;k++)
    {

C[i][j]+=A[i][k]*B[k][j];
}
}

for(i=0;i<n;i++)
{
    for(j=0;j<p;j++)
{
printf("%d ", C[i][j]);
}
printf("\n");
}
system("PAUSE");
return 0;
```

Hvala na pažnji!