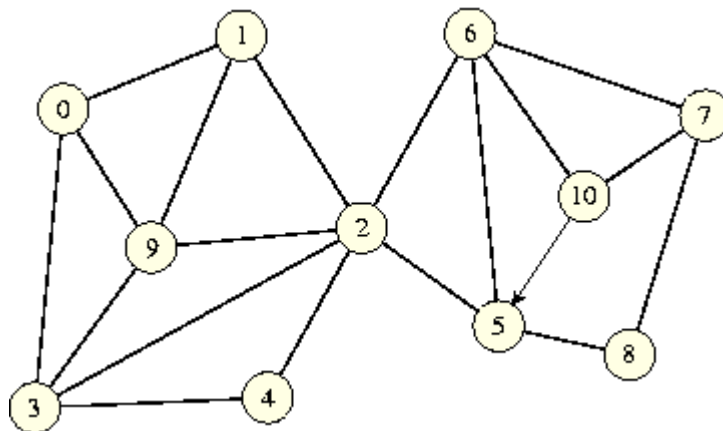


Iz matrice susedstva grafa G, formirati matricu rastojanja.

Graf G:



matrica rastojanja grafa G:

0	1	2	1	2	3	3	4	4	1	4
1	0	1	2	2	2	2	3	3	1	3
2	1	0	1	1	1	1	2	2	1	2
1	2	1	0	1	2	2	3	3	1	3
2	2	1	1	0	2	2	3	3	2	3
3	2	1	2	2	0	1	2	1	2	1
3	2	1	2	2	1	0	1	2	2	1
4	3	2	3	3	2	1	0	1	3	1
4	3	2	3	3	1	2	1	0	3	2
1	1	1	1	2	2	2	3	3	0	3
4	3	2	3	3	1	1	1	2	3	0

Program u C kodu:

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

```
struct matrica
{
    int mat[20][20];
};
```

```
void ispitaj(int a[20][20], struct matrica *Niz, int n, int d[20][20])
{
    int i,j,k,l,m;

    for(i = 0; i < n; i++)
```

```

        for(j = 0; j < n; j++)
        {
            if (i == j)    continue;
            k=1;
            if(a[i][j] == 0)
                ispitaj2(Niz,n,j,i,k,d);
            else d[i][j] = k;
        }
    }

```

```

void ispitaj2(struct matrica *Niz,int n,int j,int i,int k,int d[20][20])
{
    int l,m;
    int a[20][20];

    for(l = 0; l < n; l++)
        for(m = 0; m < n; m++)
            a[l][m] = Niz[k].mat[l][m];
    if (a[i][j] == 0)
    {
        k++;
        ispitaj2(Niz,n,j,i,k,d);
    }
    else d[i][j] = k + 1;
}

```

```

void mnozi(int a[20][20],int c[20][20],int aa[20][20], int n)
{
    int i,j,k;
    for (i = 0; i < n; i++)
        for (j = 0; j < n; j++)
        {
            c[i][j] = 0;
            for (k = 0; k < n; k++)
                c[i][j] += aa[i][k] * a[k][j];
        }
    return;
}

```

```

void form(int d[20][20],int n)
{
    int i,j;
    d[0][0] = 0;
    for (i = 0; i < n; i++)
        for (j = 0; j < n; j++)
            if (i == j)
                d[i][j] = 0;
}

```

```

void stampaj(int d[20][20], int n)
{
    int i,j;
    for (i = 0; i < n; i++)
    {
        for (j = 0; j < n; j++)
            printf("%d\t",d[i][j]);
        printf("\n");
    }
}

void formniz(int a[20][20], int c[20][20], int n, struct matrica Niz[20])
{
    int aa[20][20],i,j,k;
    for(i = 0; i < n; i++)
        for(j = 0; j < n; j++)
        {
            aa[i][j] = a[i][j];
            Niz[0].mat[i][j] = a[i][j];
        }
    stampaj(a,n);
    printf("\n");
    for(i=1;i<n-1;i++)
    {
        mnozi(a,c,aa,n);
        stampaj(c,n);
        printf("\n");
        for(j=0;j<n;j++)
            for(k=0;k<n;k++)
            {
                aa[j][k] = c[j][k];
                Niz[i].mat[j][k] = c[j][k];
            }
    }
}

main()
{
    int n,i,j;
    int a[20][20],c[20][20];
    int d[20][20];
    struct matrica Niz[20];

    printf("Unesite dimenziju matrice susedstva n:\n");scanf("%d",&n);
    for(i = 0; i < n; i++)
        for(j = 0; j < n; j++)
        {
            printf("\nUnesite %d. clan %d. vrste:\t",j+1,i+1);
            scanf("%d",&a[i][j]);
        }
}

```

```

        d[i][j] = 0;
    }
    formniz(a,c,n,Niz);
    ispitaj(a,Niz,n,d);
    form(d,n);
    printf("\n");
    stampaj(d,n);
}

```

OBJASNJENJE:

Na pocetku unesemo matricu susedstva grafa G,a m-cu rastojanja d postavimo da svuda stoje nule. Pozivamo funkciju **formniz** koja pravi niz matrica setnji grafa G,tj n – 1 grana moze maksimalno da ucestvuje u putu od jednog do drugog cvora.

Napomena: Matrica rastojanja podrazumeva najkraca rastojanja od jednog do drugog cvora, dok se za isti cvor podrazumevano stavlja 0!

Zatim funkcija **ispitaj** proverava prvo u matrici susedstva svaki broj koji nije na dijagonali da li je razlicit od 0,tj nije jednak 1. Za poziciju a[i][j] gde je 0 poziva se funkcija **ispitaj2** koja iz niza matrica setnji uzima prvo matricu a²I ispituje da li se I tu nalazi 0,ukoliko se nadje, **ispitaj2** se rekurzivno poziva sve dok se ne nadje setnja koja postoji izmedju trazениh cvorova, tj a[i][j] != 0.

form postavlja 0 na glavnu dijagonalu.

I na **stampaj** ispisuje trazenu matricu rastojanja.