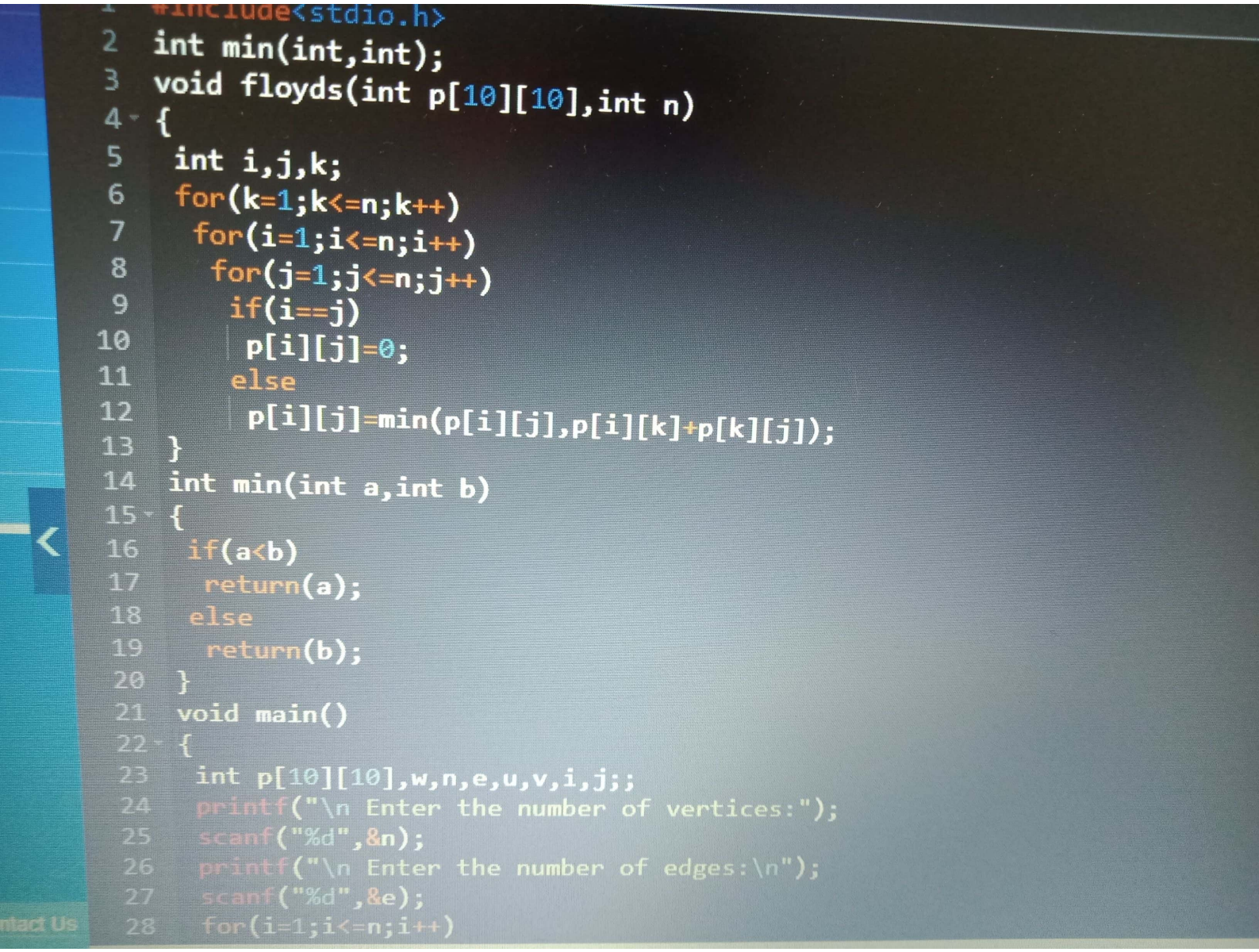
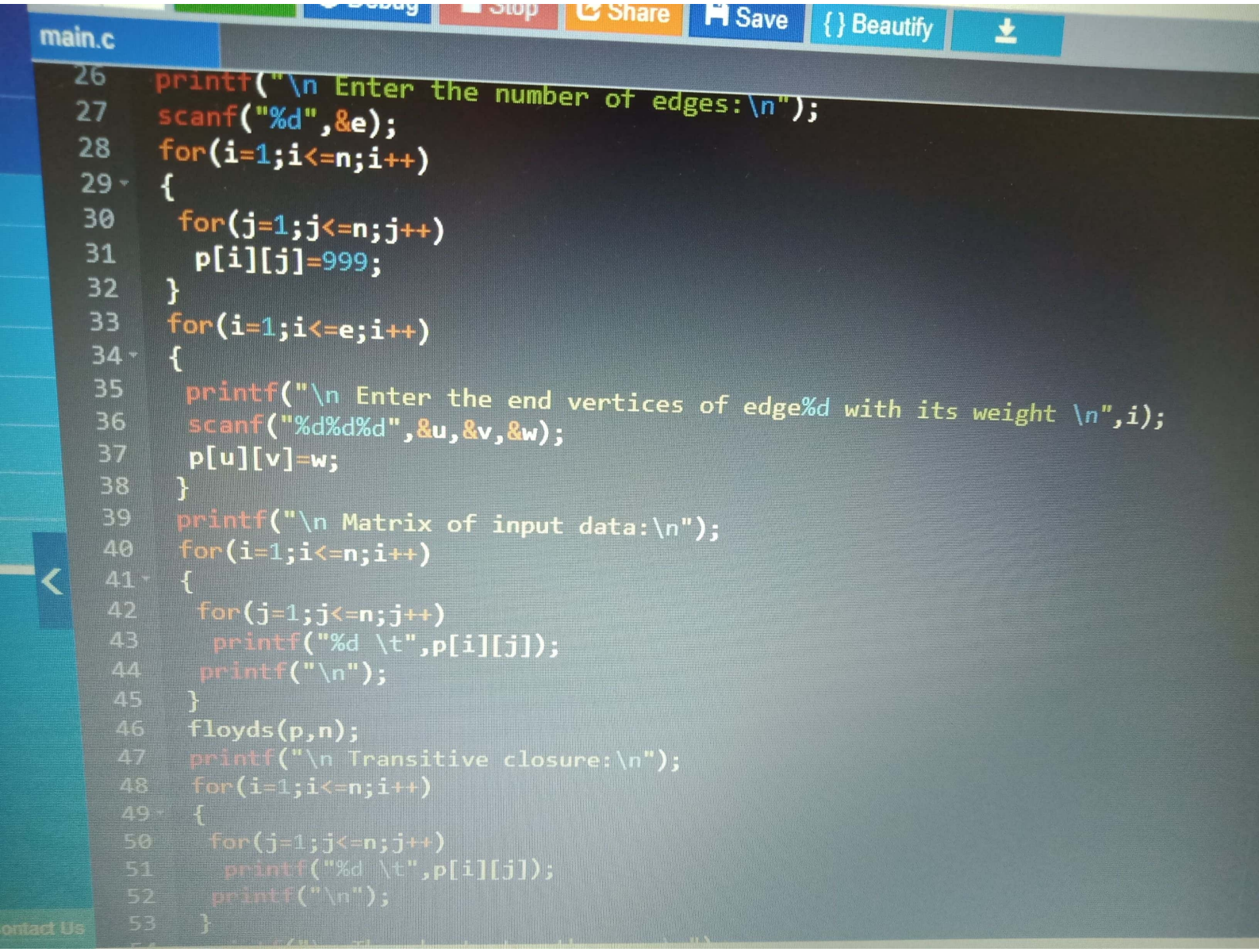


```c
#include<stdio.h>
int min(int,int);
void floyds(int p[10][10],int n)
{
 int i,j,k;
 for(k=1;k<=n;k++)
  for(i=1;i<=n;i++)
   for(j=1;j<=n;j++)
    if(i==j)
     p[i][j]=0;
    else
     p[i][j]=min(p[i][j],p[i][k]+p[k][j]);
}
int min(int a,int b)
{
 if(a<b)
  return(a);
 else
  return(b);
}
void main()
{
 int p[10][10],w,n,e,u,v,i,j;;
 printf("\n Enter the number of vertices:");
 scanf("%d",&n);
 printf("\n Enter the number of edges:\n");
 scanf("%d",&e);
 for(i=1;i<=n;i++)
```



```
35    printf("\n Enter the end vertices of edge%d with its weight \
36    scanf("%d%d%d",&u,&v,&w);
37    p[u][v]=w;
38   }
39   printf("\n Matrix of input data:\n");
40   for(i=1;i<=n;i++)
41   {
42    for(j=1;j<=n;j++)
43     printf("%d \t",p[i][j]);
44    printf("\n");
45   }
46   floyds(p,n);
47   printf("\n Transitive closure:\n");
48   for(i=1;i<=n;i++)
49   {
50    for(j=1;j<=n;j++)
51     printf("%d \t",p[i][j]);
52    printf("\n");
53   }
54   printf("\n The shortest paths are:\n");
55   for(i=1;i<=n;i++)
56    for(j=1;j<=n;j++)
57    {
58     if(i!=j)
59      printf("\n <%d,%d>=%d",i,j,p[i][j]);
60    }
61  }
```

```
Enter the number of vertices:4

Enter the number of edges:
5

Enter the end vertices of edge1 with its weight
1 2 10

Enter the end vertices of edge2 with its weight
3 4 20

Enter the end vertices of edge3 with its weight
2 3 15

Enter the end vertices of edge4 with its weight
4 5 40

Enter the end vertices of edge5 with its weight
1 3 50

Matrix of input data:
999     10      50      999
999     999     15      999
999     999     999     20
999     999     999     999

Transitive closure:
0       10      25      45
999     0       15      35
999     999     0       20
999     999     999     0
```