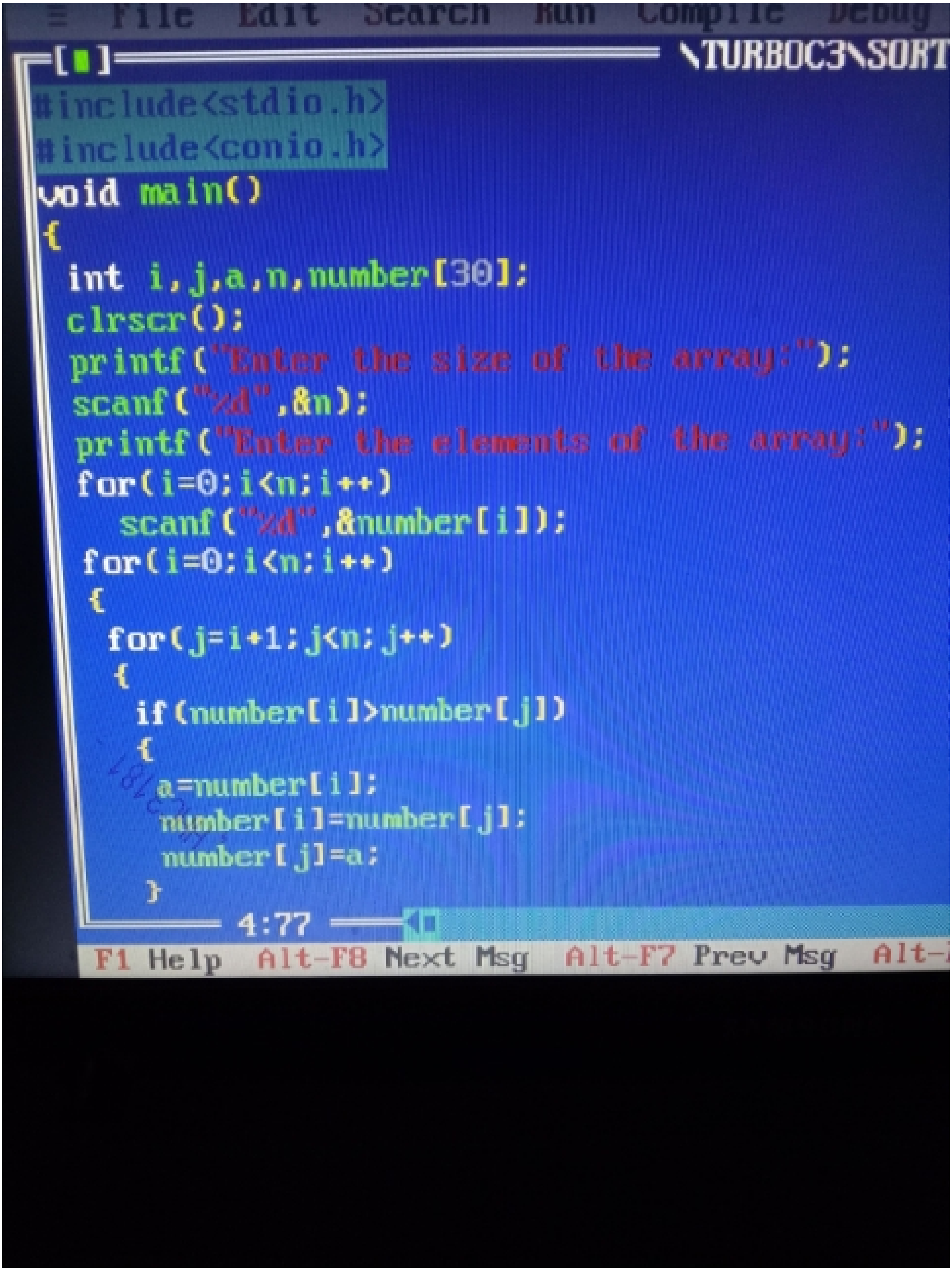
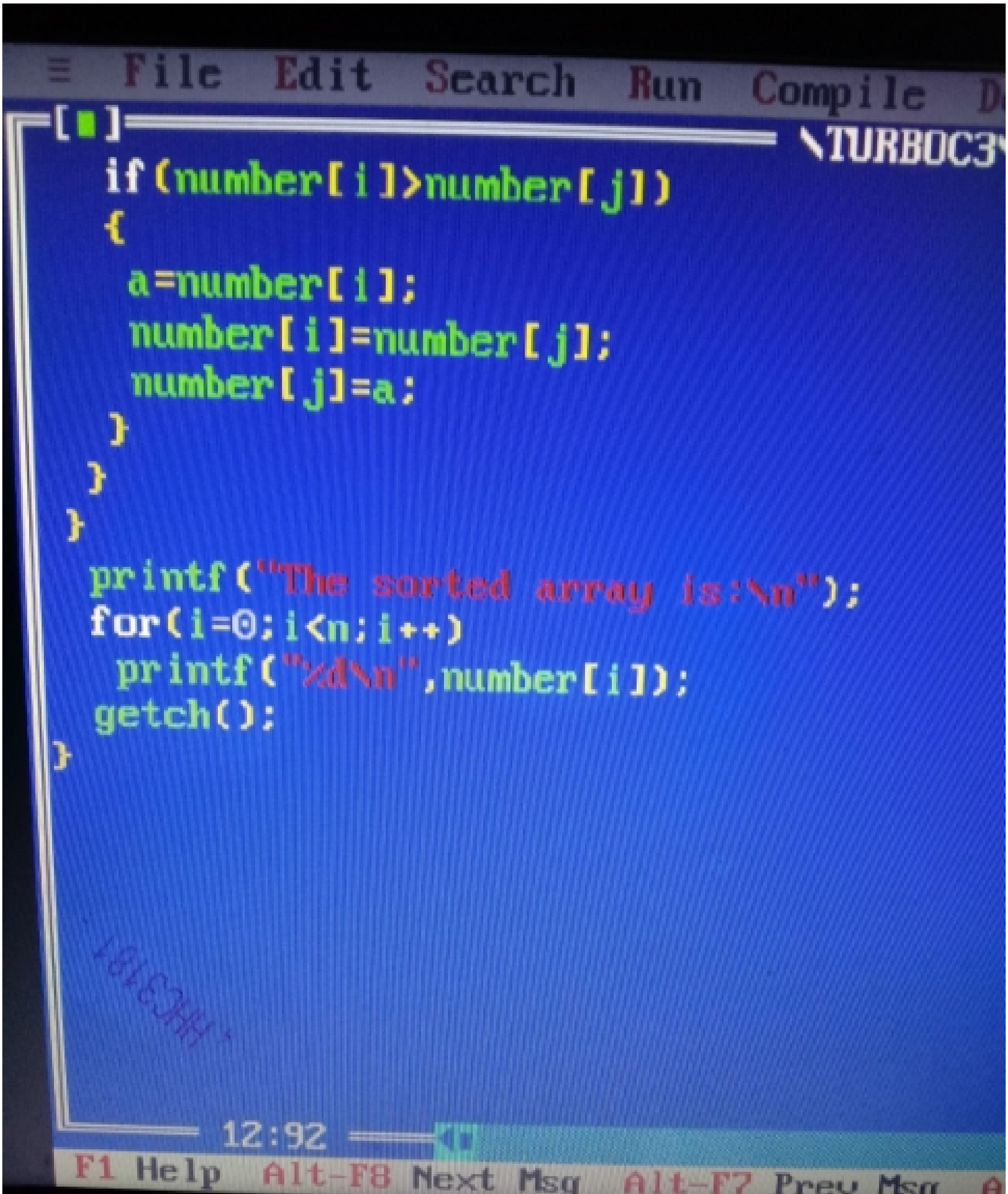
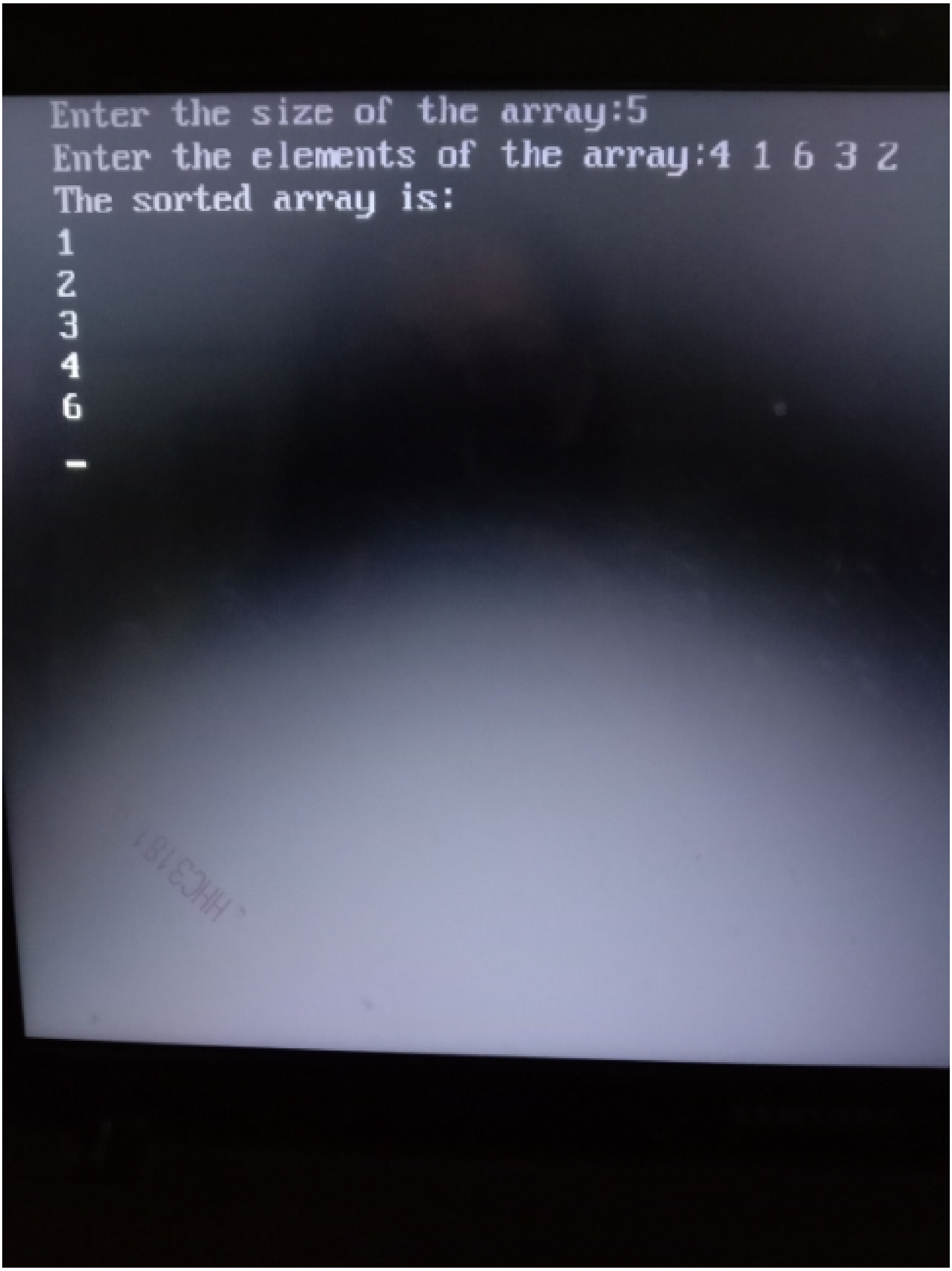

≡ File Edit Search Run Compile Debug
[■]
\TURBOC3\SORT
4:77
F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt-

```
#include<stdio.h>
#include<conio.h>
void main()
{
 int i,j,a,n,number[30];
 clrscr();
 printf("Enter the size of the array:");
 scanf("%d",&n);
 printf("Enter the elements of the array:");
 for(i=0;i<n;i++)
   scanf("%d",&number[i]);
 for(i=0;i<n;i++)
 {
  for(j=i+1;j<n;j++)
  {
   if(number[i]>number[j])
   {
    a=number[i];
    number[i]=number[j];
    number[j]=a;
   }
```

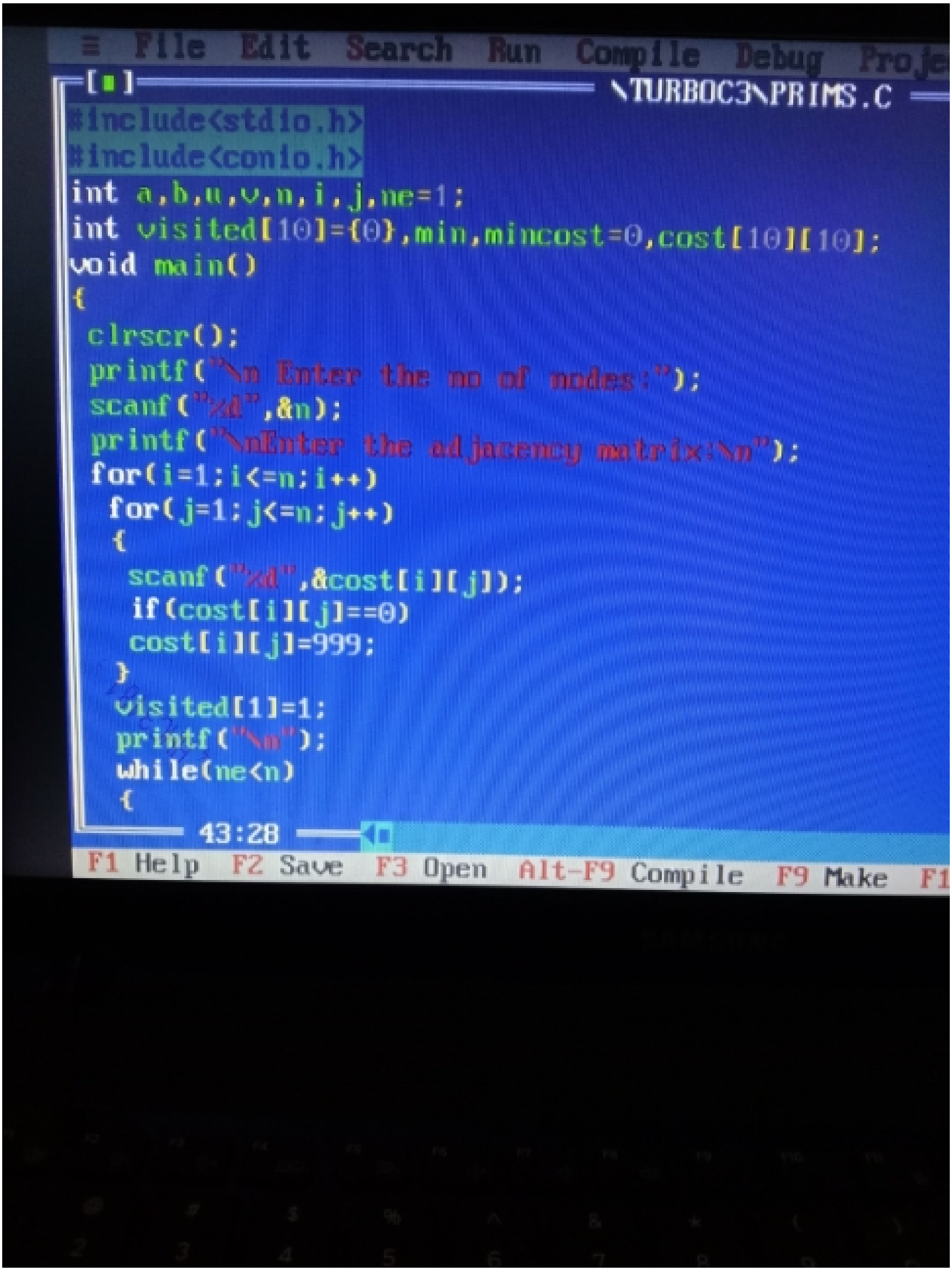


≡ File Edit Search Run Compile D
[■]
\TURBOC3
12:92
F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg A

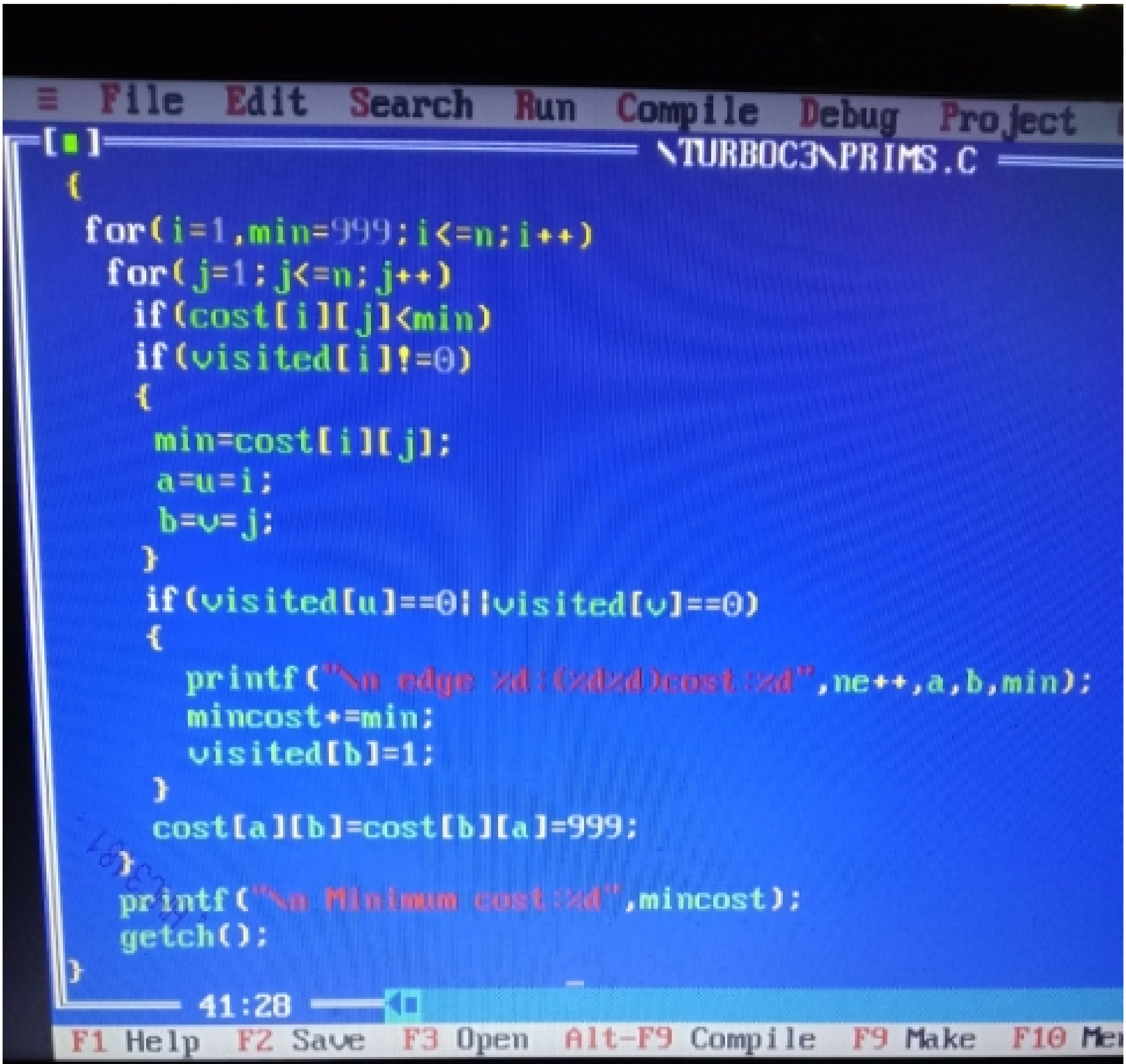
```
   if(number[i]>number[j])
   {
    a=number[i];
    number[i]=number[j];
    number[j]=a;
   }
  }
 }
 printf("The sorted array is:\n");
 for(i=0;i<n;i++)
  printf("%d\n",number[i]);
 getch();
}
```



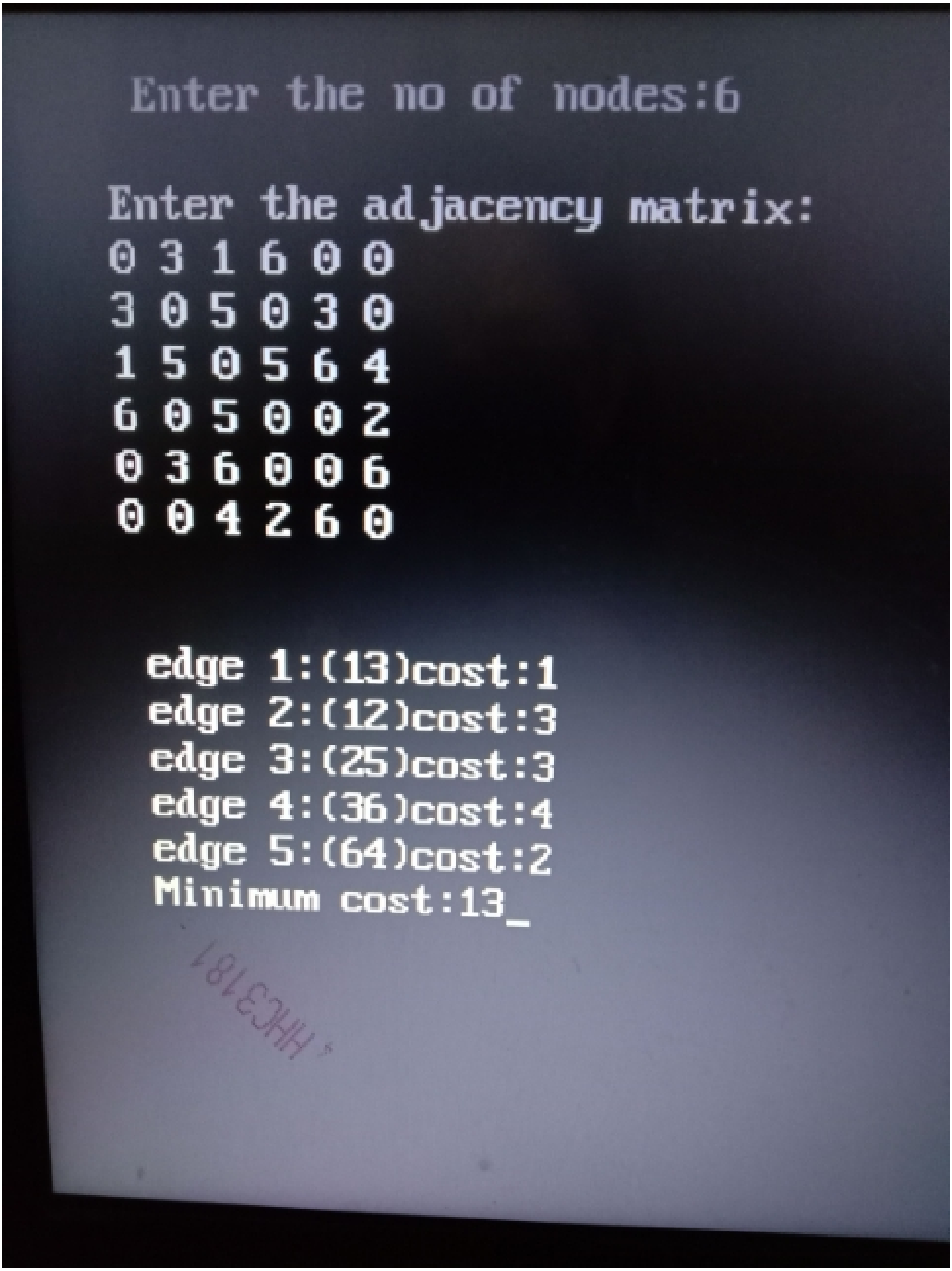
```
Enter the size of the array:5
Enter the elements of the array:4 1 6 3 2
The sorted array is:
1
2
3
4
6
_
```


≡ File Edit Search Run Compile Debug Proje
[■]
\TURBOC3\PRIMS.C
43:28
F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F1

```
#include<stdio.h>
#include<conio.h>
int a,b,u,v,n,i,j,ne=1;
int visited[10]={0},min,mincost=0,cost[10][10];
void main()
{
 clrscr();
 printf("\n Enter the no of nodes:");
 scanf("%d",&n);
 printf("\nEnter the adjacency matrix:\n");
 for(i=1;i<=n;i++)
  for(j=1;j<=n;j++)
  {
   scanf("%d",&cost[i][j]);
   if(cost[i][j]==0)
   cost[i][j]=999;
  }
  visited[1]=1;
  printf("\n");
  while(ne<n)
  {
```


≡ File Edit Search Run Compile Debug Project
[■]
\TURBOC3\PRIMS.C
41:28
F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Me

```
 {
  for(i=1,min=999;i<=n;i++)
   for(j=1;j<=n;j++)
    if(cost[i][j]<min)
    if(visited[i]!=0)
    {
     min=cost[i][j];
     a=u=i;
     b=v=j;
    }
    if(visited[u]==0||visited[v]==0)
    {
      printf("\n edge %d:(%d%d)cost:%d",ne++,a,b,min);
      mincost+=min;
      visited[b]=1;
    }
    cost[a][b]=cost[b][a]=999;
  }
  printf("\n Minimum cost:%d",mincost);
  getch();
}
```



```
Enter the no of nodes:6

Enter the adjacency matrix:
0 3 1 6 0 0
3 0 5 0 3 0
1 5 0 5 6 4
6 0 5 0 0 2
0 3 6 0 0 6
0 0 4 2 6 0


 edge 1:(13)cost:1
 edge 2:(12)cost:3
 edge 3:(25)cost:3
 edge 4:(36)cost:4
 edge 5:(64)cost:2
 Minimum cost:13_
```