shortest_nodetonode

```
#include<stdio.h>
int main()
{
    int i,j,k,ar1[100][100],ar2[100][100],n,x,y;
    printf("Enter the number of vertices\n");
    scanf("%d",&n);

    printf("Enter the elements of the array\n");
    for(i=0;i<n;i++){
        for(j=0;j<n;j++){
            scanf("%d",&ar1[i][j]);
        }
    }

    printf("Enter the vertices1:");
    scanf("%d",&x);
    printf("\nEnter the vertices2:");
    scanf("%d",&y);

    for(i=0;i<n;i++){
        for(j=0;j<n;j++){
            if(ar1[i][j]>=1)
            {
                ar2[i][j]=ar1[i][j];
            }
            else
            {
                ar2[i][j]=9999;
            }
        }
    }

    for(k=0;k<n;k++){
        for(i=0;i<n;i++){
            for(j=0;j<n;j++){
               if( ar2[i][j]<(ar2[i][k]+ar2[k][j]))
               {
                   ar2[i][j]=ar2[i][j];
               }
               else
               {
                   ar2[i][j]=(ar2[i][k]+ar2[k][j]);
               }
            }
        }
    }
    printf(" /nshortest path length from %d to %d is: %d\n",x,y,ar2[x-1][y-1]);
    return 0;

}
```

```
--------------------------------------------------------------------------------------------------------------------------------
-----------------------
```

selectionkotochaw

```
#include<stdio.h>
int main()
{
int i,j,n,a[100],temp=0,x,y;
printf("Enter array size:");
scanf("%d",&n);
printf("Enter array element:");
for(i=0;i<n;i++){
   scanf("%d",&a[i]);
}
for(i=0;i<n;i++){
    for(j=0;j<n;j++){
       if(a[i]<a[j]){
        temp=a[i];
        a[i]=a[j];
        a[j]=temp;
       }
    }
}
printf(" \narray element: ");
for(i=0;i<n;i++){
   printf("%d ",a[i]);
}

printf("Enter m kotogula chaw:");
scanf("%d",&x);
```

```
for(i=0;i<x;i++){
printf("Enter data:");
scanf("%d",&y);
a[n+i]=y;
}

 for(i=0;i<n+x;i++){
    for(j=0;j<n+x;j++){
       if(a[i]<a[j]){
        temp=a[i];
        a[i]=a[j];
        a[j]=temp;
       }

    }

}
printf(" \narray element: ");
for(i=0;i<n+x;i++){
   printf("%d ",a[i]);
}

return 0;
}





------------------------------------------------------------------------------------------------------------------------------------
-----------------------
selection(1)









#include<stdio.h>

int main()
{
int i,j,n,a[100],temp=0;
printf("Enter array size:");
scanf("%d",&n);
printf("Enter array element:");
for(i=0;i<n;i++){
   scanf("%d",&a[i]);
}

 for(i=0;i<n;i++){
    for(j=0;j<n;j++){
       if(a[i]<a[j]){
        temp=a[i];
        a[i]=a[j];
        a[j]=temp;
       }
    }
}
printf(" \nSorted Array Element: ");
for(i=0;i<n;i++){
   printf("%d ",a[i]);
}

return 0;
}
```

```
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
----------------------------------
quicksort
```

```
#include<stdio.h>
int partition(int arr[],int lb,int ub)
{
    int i,j,pi,temp;
    i=lb-1;
    pi=arr[ub];
    for(j=lb;j<ub;j++)
    {
        if(pi>arr[j])
        {
            i++;
            temp=arr[i];
            arr[i]=arr[j];
            arr[j]=temp;
        }
    }
    i++;
    temp=arr[i];
    arr[i]=pi;
    arr[ub]=temp;
    return i;
```

```
}
void qc(int arr[],int lb,int ub)
{
    int pi;
    if(lb<ub)
    {
        pi=partition(arr,lb,ub);
        qc(arr,lb,pi-1);
        qc(arr,pi+1,ub);
    }
}
main()
{
    int n,i,arr[100];
    printf("Enter array size:");
    scanf("%d",&n);
    printf("Enter element in array:");
    for(i=0;i<n;i++)
    scanf("%d",&arr[i]);

    qc(arr,0,n-1);
    printf("Array after sorting:");
    for(i=0;i<n;i++)
        printf("%d\t",arr[i]);
}


------------------------------------------------------------------------------------------------------------------------------------------
-------------------------
path_matrix











#include<stdio.h>

int main()
{
    int n,i,j,k,ar1[100][100],ar2[100][100];
    printf("Enter the number of vertices: ");
    scanf("%d",&n);

    printf("Enter elements: \n");
    for(i=0;i<n;i++){
        for(j=0;j<n;j++){
            scanf("%d",&ar1[i][j]);
        }
    }
    for(i=0;i<n;i++){
        for(j=0;j<n;j++){
            if(ar1[i][j]>=1){
                ar2[i][j]=1;
            }
            else{
                ar2[i][j]=0;
            }
        }
    }
    for(k=0;k<n;k++){
     for(i=0;i<n;i++){
       for(j=0;j<n;j++){
            ar2[i][j]=ar2[i][j]||(ar2[i][k]&&ar2[k][j]);
        }
    }
}

printf("\nPath Matrix: \n");

    for(i=0;i<n;i++){
        for(j=0;j<n;j++){
            printf("%d ",ar2[i][j]);
        }
     printf("\n");
    }

return 0;
}




------------------------------------------------------------------------------------------------------------------------------------
-----------------------
insertionkotogola









#include<stdio.h>
int main(){

int i,j,n,a[100],temp=0,x,y;
printf("Enter array size:");
scanf("%d",&n);
printf("Enter array element:");
for(i=0;i<n;i++){
   scanf("%d",&a[i]);
}
```

```
  for(i=1;i<n;i++){
    for(j=i;j>0;--j){
       if(a[j]<a[j-1]){
        temp=a[j];
        a[j]=a[j-1];
        a[j-1]=temp;
       }
       else{
        break;
       }
    }
}
printf(" \narray element:");
for(i=0;i<n;i++){
   printf("%d ",a[i]);
}

printf("Enter m kotogula chaw:");
scanf("%d",&x);
for(i=0;i<x;i++){
printf("Enter data:");
scanf("%d",&y);
a[n+i]=y;
}


  for(i=1;i<n+x;i++){
    for(j=i;j>0;--j){
       if(a[j]<a[j-1]){
        temp=a[j];
        a[j]=a[j-1];
        a[j-1]=temp;
       }
       else{
        break;
       }
    }
}
printf(" \nLatest Array Element:");
for(i=0;i<n+x;i++){
   printf("%d ",a[i]);
}

return 0;
}




-----------------------------------------------------------------------------------------------------------------------------------------------------------------------
insertion (1)
```

```
#include<stdio.h>

int main(){

int n,i,j,a[100],temp=0;
printf("Enter number of array:");
scanf("%d",&n);
printf("Enter the array element:");
for(i=0;i<n;i++){
scanf("%d",&a[i]);
}

for(i=1;i<n;i++){
    for(j=i;j>0;--j){
        if(a[j]<a[j-1]){
        temp=a[j];
        a[j]=a[j-1];
        a[j-1]=temp;
        }
       else{
        break;
        }
    }
}
printf(" \narray element:");
for(i=0;i<n;i++){
   printf("%d",a[i]);
}

return 0;
}
```


Zaki_Path_Matrix Warshal's

```
 //Warshall’s algorithm to find the path matrix.


#include <stdio.h>
#include <conio.h>


void read (int mat[5][5], int n);
void display (int mat[5][5], int n);
void mul(int mat[5][5], int n);


void main()
{
int adj[5][5], P[5][5], n, i, j, k;

printf("\n Enter the number of nodes in the graph : ");
scanf("%d", &n);
printf("\n Enter the adjacency matrix : ");
read(adj, n);

printf("\n The adjacency matrix is : ");
display(adj, n);
for(i=0;i<n;i++)
{
for(j=0;j<n;j++)
{
if(adj[i][j] == 0)
P[i][j] = 0;
else
P[i][j] = 1;
}
}
for(k=0; k<n;k++)
{
for(i=0;i<n;i++)
{
for(j=0;j<n;j++)
P[i][j] = P[i][j] | ( P[i][k] & P[k][j]);
}
}
printf("\n The Path Matrix is :");
display (P, n);
}


void read(int mat[5][5], int n)
{
int i, j;
for(i=0;i<n;i++)
{
for(j=0;j<n;j++)
{
printf("\n mat[%d][%d] = ", i, j);
scanf("%d", &mat[i][j]);
}
}
}


void display(int mat[5][5], int n)
{
int i, j;
for(i=0;i<n;i++){
    printf("\n");
for(j=0;j<n;j++)
printf("%d\t", mat[i][j]);
}

}
```


Zaki_Shortest_Path

```
#include<stdio.h>
main()
{
    int n,arr[100][100],p[100][100],i,j,k;
    printf("Enter vertix number:");
    scanf("%d",&n);
    printf("Enter matrix:");
    for(i=0;i<n;i++)
        for(j=0;j<n;j++)
        scanf("%d",&arr[i][j]);
    for(i=0;i<n;i++)
        for(j=0;j<n;j++){
        if(arr[i][j]==0)
            p[i][j]=444444;
            else
                p[i][j]=arr[i][j];
        }
        for(k=0;k<n;k++)
           for(i=0;i<n;i++)
        for(j=0;j<n;j++){
                p[i][j]=(p[i][j]<(p[i][k]+p[k][j]))?p[i][j]:(p[i][k]+p[k][j]);
        }
```

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

    }

}
```

