**Exercise 12: Implementation of sorting of numbers using selection and shell sort**

**adt.h:**

#include<stdio.h>

struct num

{

int A[100];

int n;

};

void disp(struct num\*);

void init(struct num\*,int );

void ins(struct num\*,int[]);

void selSort(struct num\*);

void shellSort(struct num\*);

**impl.h:**

#include "adt.h"

void init(struct num\* N,int n)

{

N->n=n;

}

void ins(struct num\* N, int x[])

{

for(int i=0;i<N->n;i++)

{

N->A[i]=x[i];

}

}

void disp(struct num\* N)

{

for(int i=0;i<N->n;i++)

{

printf("%d ",N->A[i]);

}

}

void selSort(struct num \*N)

{

int temp,min\_i;

for(int i=0;i<N->n-1;i++)

{

min\_i=i;

for(int j=i;j<N->n;j++)

{

if(N->A[j]<N->A[min\_i])

min\_i=j;

}

temp=N->A[i];

N->A[i]=N->A[min\_i];

N->A[min\_i]=temp;

}

}

void shellSort(struct num \*N)

{

for(int gap=N->n/2;gap>0;gap/=2)

{

for(int i=gap;i<N->n;i++)

{

for(int j=i-gap;j>=0;j-=gap)

{

if(N->A[j+gap]>N->A[j])

break;

else

{

int temp=N->A[j+gap];

N->A[j+gap]=N->A[j];

N->A[j]=temp;

}

}

}

}

}

**appl.c:**

#include "impl.h"

#include<stdlib.h>

int main()

{

struct graph \*G=(struct graph \*)malloc(sizeof(struct graph));

int ch;

printf("\nMenu:\n1.Create a graph\n2.Display the graph matrices\n3.Apply Floyd Warshall\n4.Display Path\n5.Exit\n");

do

{

printf("\nChoice: ");

scanf("%d",&ch);

switch(ch)

{

case 1:

printf("Enter the number of verties and edges: ");

int v,e;

scanf("%d%d",&v,&e);

create(G,v,e,1);

break;

case 2:

disp(G);

break;

case 3:

floyd(G);

break;

case 4:

printf("Enter source and destination: ");

int s,d;

scanf("%d%d",&s,&d);

printf("\nPath from %d to %d: %d",s,d,s);

path(G,s-1,d-1);

printf("\nShoretst dist: %d\n",G->dist[s-1][d-1]);

break;

case 5:printf("\nExiting...\n");

break;

default:printf("\nInavlid Input!\n");

}

}while(ch!=5);

//create(G,3,5,1);

//int s=4,d=1;

}

**Sample I/O:**

