Task 1

#include<iostream>

using namespace std;

void function(int a, int b, int n){

if(n==1){

return;

}else{

int c = a+b;

cout<<c<<" ";

function(b,c,n-1);

}

}

int main(){

int a=0;

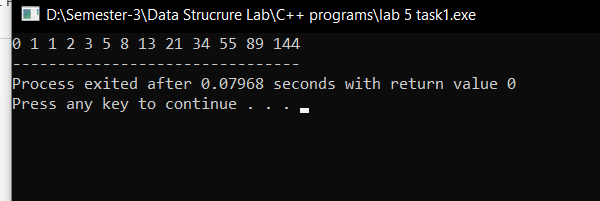
int b=1;

int n=14;

cout<<a<<" "<<b<<" ";

function(a,b,n-2);

}



Task 2

#include<iostream>

using namespace std;

int m=11;

void fun2(int,int,int,int);

void fun1(int n,int a,int b,int c)

{

if(n==m)

{

cout<<a<<" "<<b<<" ";

}

if(n==0)

{

return;

}

fun2(n,a,b,c);

}

void fun2(int n,int a,int b,int c)

{

c=a+b;

cout<<c<<" ";

a=b;

b=c;

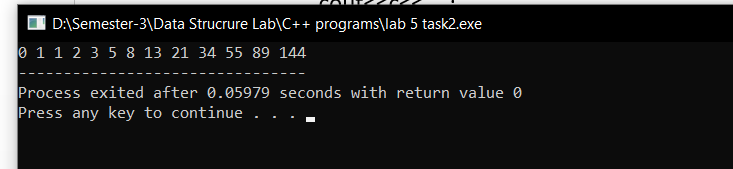
fun1(n-1,a,b,c);

}

main()

{

fun1(m,0,1,0);

}

Task 3

#include<iostream>

using namespace std;

void Print(int arr[],int n)

{

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

cout<<arr[i]<<" ";

cout<<endl;

}

void NormalSorting(int i,int j,int n,int arr[]){

if(i!=n){

if(arr[i]<arr[j])

{

swap(arr[i],arr[j]);

}

if(j==n)

NormalSorting(i+1,0,n,arr);

else

NormalSorting(i,j+1,n,arr);

}

else{

Print(arr,n);

return;

}

}

void TailSorting(int i,int j,int n,int arr1[]){

if(i==n){

Print(arr1,n);

return;

}

if(arr1[i]<arr1[j]){

swap(arr1[i],arr1[j]);

}

if(j==n)

NormalSorting(i+1,0,n,arr1);

else

NormalSorting(i,j+1,n,arr1);

}

main()

{

int arr[5]={11,13,5,6,7};

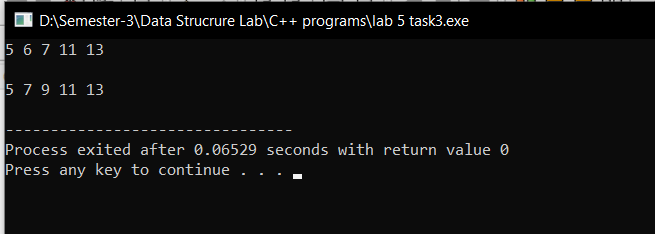
NormalSorting(0,0,sizeof(arr)/4,arr);

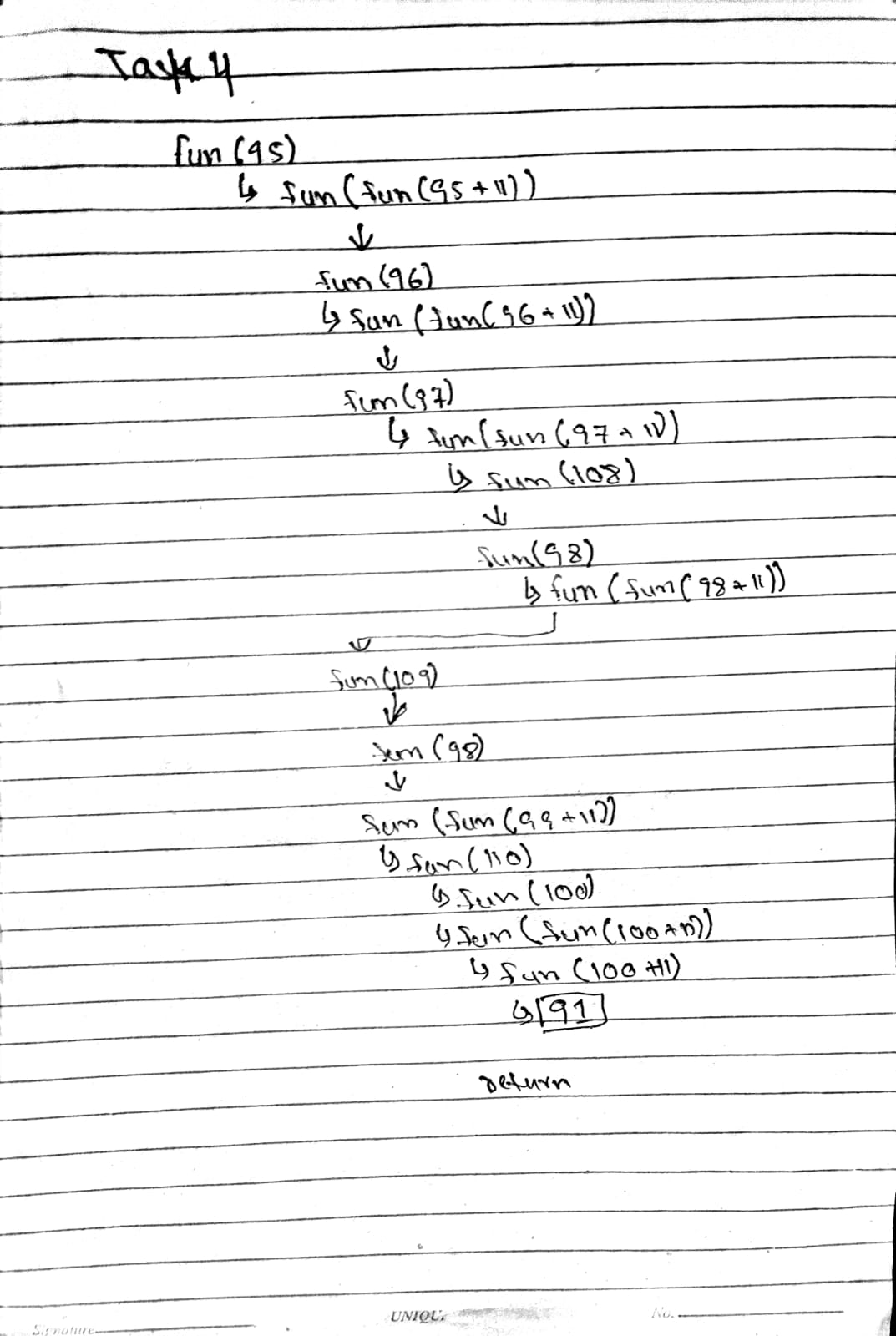
cout<<endl;

int arr1[5]={11,13,5,9,7};

TailSorting(0,0,sizeof(arr1)/4,arr1);

}





Task 5A

#include<iostream>

using namespace std;

const int n=4;

void Print(int sol[n][n]){

for(int i=0;i<n;i++) {

for(int j=0;j<n;j++){

cout<<sol[i][j]<<" ";

}

cout<<endl;

}

}

bool isSafe(int x,int y,int maze[n][n]){

if(x>=0&&x<n&&y>=0&&y<n&&maze[x][y]==1)

return true;

else

return false;

}

bool solveMaze(int maze[n][n],int x,int y,int sol[n][n]){

if(x==n-1&&y==n-1){

sol[x][y]=1;

return true;

}

if(isSafe(x,y,maze)){

sol[x][y]=1;

if(solveMaze(maze,x,y+1,sol))

return true;

if(solveMaze(maze,x+1,y,sol))

return true;

sol[x][y] = 0;

return false;

}

return false;

}

main(){

int maze[n][n]={{1,0,0,0},

{1,1,0,1},

{0,1,0,0},

{1,1,1,1}};

int sol[n][n]={0};

if (solveMaze(maze,0,0,sol))

{

cout<<"Final Path:"<<endl;

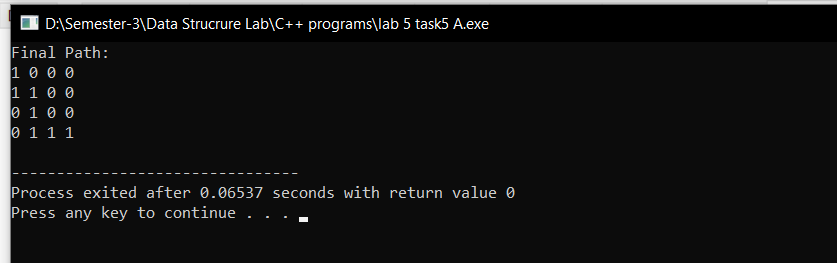
Print(sol);

}

else

cout<<"No Path Found"<<endl;

}



Task 5B

#include<iostream>

using namespace std;

const int rows=3,col=4;

void print(int sol[rows][col]){

for(int i=0;i<rows;i++)

{

for (int j=0;j<col;j++)

cout<<sol[i][j]<<" ";

cout<<endl;

}

}

bool valid(int x,int y,int maze[rows][col]){

if(x>=0&&x<rows&&y>=0&&y<col&&maze[x][y]==1)

return true;

else

return false;

}

bool findpath(int path[rows][col], int x, int y, int sol[rows][col]){

if(x==0&&y==3)

{

sol[x][y]=1;

return true;

}

if(valid(x,y,path)){

sol[x][y]=1;

if(findpath(path,x,y+1,sol))

return true;

if(findpath(path,x+1,y,sol))

return true;

sol[x][y] = 0;

return false;

}

return false;

}

int main(){

int path[3][4]={

{1,0,0,1},

{0,1,1,1},

{0,1,1,0}};

int s[3][4]={0};

if(findpath(path,0,0,s)){

cout << "Output path:" << endl;

print(s);

}

else

cout<<"Output Not found"<<endl;

}

