Name: Vishal J Lodha

Section: I

Semester: 3rd

SRN: PES1UG20CS507

Date: 15/11/2021

File Name:PES1UG20CS507\_F.c

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

#include "header.h"

int main()

{

st head;

init(&head);

FILE \*fp=fopen("input.txt","r");

int r[2];

int c[2];

read(r,2,fp);

read(c,2,fp);

create(&head,c[0]-r[0]+1,c[1]-r[1]+1,r[0],r[1],fp);

del(&head,c[0]-r[0],c[1]-r[1]);

path(&head,c[0]-r[0],c[1]-r[1]);

return 0;

}

File Name:PES1UG20CS507\_H.h

typedef struct node{

int val;

int row,col;

struct node \*r;

struct node \*d;

}no;//node definition

typedef struct start{

no \*head;

}st;//multilist definition

void init(st \*p);

no \*nod(int val);

void read(int \*a,int n,FILE \*fp);

void create(st \*p,int row,int col,int rs,int cs,FILE \*fp);

int rowdel(no \*p,int up,int col);

void del(st \*p,int row,int col);

int check(no \*p,int row,int col);

void path(st \*p,int row,int col);

File Name: PES1UG20CS507\_C.c

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include "header.h"

void init(st \*p)//initialisation the starting of multilist

{

p->head=NULL;

}

no \*nod(int val)//creating of node

{

no \*temp=(no\*)malloc(sizeof(no));

temp->val=val;

temp->r=NULL;

temp->d=NULL;

return temp;

}

void read(int \*a,int n,FILE \*fp)//reading a line of inputs that are seperated by space

{

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

{

fscanf(fp,"%d",a+i);

}

}

void create(st \*p,int row,int col,int rs,int cs,FILE \*fp)//creating a multilist from starting point to ending point

{

int arr[col+cs];

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

read(arr,col+cs,fp);

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

{

no \*q=p->head;

no \*temp=nod(arr[i+cs]);

temp->row=1+rs;

temp->col=i+1+cs;

if(q==NULL)

{

p->head=temp;

}

else

{

while(q->r!=NULL)

{

q=q->r;

}

q->r=temp;

}

}

for(int i=0;i<(row-1);i++)

{

no \*q=p->head;

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

{

q=q->d;

}

no \*m=NULL;

int c=1;

read(arr,col+cs,fp);

while(q!=NULL)

{

no \*temp=nod(arr[c-1+cs]);

temp->row=i+2+rs;

temp->col=c+cs;

c++;

q->d=temp;

if(m!=NULL)

{

m->d->r=temp;

}

m=q;

q=q->r;

}

}

}

int rowdel(no \*p,int up,int col)//deleting nodes if they are usless or unreachable in row form

{

no \*q=NULL;

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

{

if(i<up)

{

q=p;

p=p->r;

}

else

{

if(p->val==0)

{

up=i;

q=p;

p=p->r;

}

else

{

q->r=NULL;

return up;

}

}

}

return up;

}

void del(st \*p,int row,int col)//deleting all nodes that cannot be reached

{

int up=0;

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

{

no \*q=p->head;

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

{

q=q->d;

}

up=rowdel(q,up,col);

}

}

int check(no \*p,int row,int col)//checks if by going to the next node they can reach the destination

{

int right=0;

int down=0;

if(p->r!=NULL && p->d!=NULL)

{

if(p->r->val==1 && p->d->val==1)

return 0;

}

if(p->row==row && p->col==col)

return 1;

if(p->r!=NULL)

{

if(p->val==0)

{

if(p->r->val==0)

right=check(p->r,row,col);

}

}

if(p->d!=NULL)

{

if(p->val==0)

{

if(p->d->val==0)

down=check(p->d,row,col);

}

}

if(right==1 || down==1)

return 1;

return 0;

}

void path(st \*p,int row,int col)//prints the output in the output file

{

FILE \*fp=fopen("out.txt","w");

no \*q=p->head;

while(1)

{

fprintf(fp,"%d,%d\n",q->row-1,q->col-1);

int ctr=0;

if(q->r!=NULL)

{

if(check(q->r,row,col))

{

q=q->r;

ctr=1;

}

}

if(q->d!=NULL && ctr==0)

{

if(check(q->d,row,col))

{

q=q->d;

ctr=1;

}

}

if(ctr==0)

break;

}

if(q->row==row && q->col==col)

{

if(q->r->val==0)

{

fprintf(fp,"%d,%d\n%d,%d\n",q->r->row-1,q->r->col-1,row,col);

}

else if(q->d->val==0)

{

fprintf(fp,"%d,%d\n%d,%d\n",q->d->row-1,q->d->col-1,row,col);

}

else

printf("No path to reach\n");

}

fclose(fp);

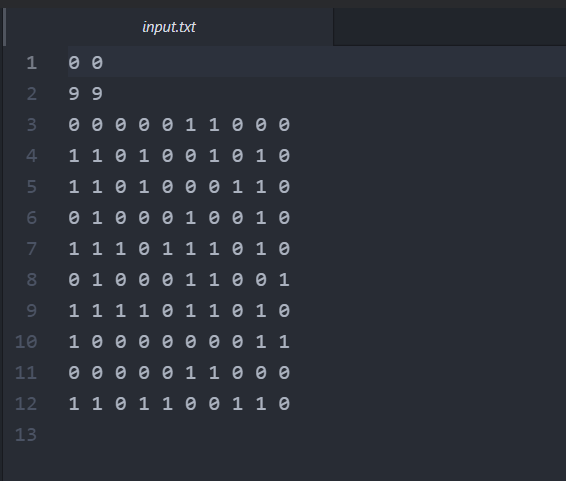
}

Output Screenshots:

Command Promt:



Input File:



Output File:

