#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#include <stdbool.h>

#include <string.h>

#include <conio.h>

/\*Declaring struct\*/

typedef struct{

char name[1000];

}fname;

/\*Read from file function\*/

void readFile(){

FILE \*fp;

int c;

fp=fopen("score.txt","r");

while(1)

{

c=fgetc(fp);

if(feof(fp))

{

break;

}

printf("%c",c);

}

fclose(fp);

}

/\*Write to file function\*/

void writeFile(char input[1000], int score)

{

int size;

FILE \*fp;

fp=fopen("score.txt","r");

if(NULL !=fp)

{

fseek(fp,0,SEEK\_END);

size=ftell(fp);

if(size==0)

{

fp=fopen("score.txt","w");

fprintf(fp,"%s\t\t%d\n",input,score);

fclose(fp);

}

else

{

fp=fopen("score.txt","a");

fprintf(fp,"%s\t\t%d\n",input,score);

fclose(fp);

}

}

}

/\*START SCREEN function\*/

void startScreen(fname \*start)

{

char input[1000];

printf("Welcome to the Manhattan Tourist Problem!\n");

printf("\nPrevious Player Records\n");

printf("Name\t\tScore(attractions visited)\n");

readFile(); /\*Display previous player records\*/

printf("\nPlease key in your name please:");

scanf("%s",&input);

strcpy((\*start).name,input);

fflush(stdin);

printf("\nGood day %s,let's start the game...all the best!!!\n",(\*start).name);

printf("Press any key to start the game...\n");

\_getch(); /\*Press any key to move on\*/

system("cls");

}

/\*Manhattan Tourist Problem function\*/

int grid(){

int matrix[15][8]; /\*Creation of grid\*/

int row,col;

int row\_sel[250]={0,14},col\_sel[250]={0,7}; /\*Array used to store past selection\*/

int a;

bool foundMatch;

int row\_box=0,col\_box=0,score=0; /\*Used to keep track of position in grid\*/

int temp=1;

char input;

int loop1=0;

srand((unsigned)time(NULL)); /\*Seeding the grid based on time\*/

for(row=0;row<15;row++) /\*Initializing values in the grid\*/

{

for(col=0;col<8;col++)

{

if(row==0 && col==0)

{

matrix[row][col]=0;

}

else

{

matrix[row][col]=rand()%10;

}

}

}

while(loop1!=1)

{

for(row=0;row<15;row++)

{

for(col=0;col<8;col++)

{

foundMatch=false; /\*This is to track previous movement, point(0,0) and point(14,7) are selected at the beginning.\*/

for(a=0;!foundMatch && a<sizeof(row\_sel)/(sizeof(row\_sel[0]));a++) /\*condition is foundMatch is TRUE AND a less than array size\*/

{

foundMatch=row==row\_sel[a] && col==col\_sel[a]; /\*foundMatch is assigned boolean of condition, returns 0 if true and 1 if false\*/

}

if(foundMatch)/\*if 1 because 1 means false\*/

{

if(col==7)

{

printf("-|%d|-",matrix[row][col]);

}

else

{

printf("-|%d|",matrix[row][col]);

}

}

else

{

if(row==1||row==3||row==5||row==7||row==9||row==11||row==13)

{

printf(" | ");

}

else

{

if(col==7)

{

printf("- %d -",matrix[row][col]);

}

else

{

printf("- %d ",matrix[row][col]);

}

}

}

}

printf("\n");

}

printf("Number of attractions visisted so far:%d\n\n",score);

printf("Each number in the map represents the number of attractions along each street.\n\n");

printf("You may only move to the right or move down!\n\n");

printf("Press 'R' to go right\n");

printf("Press 'D' to move down\n");

printf("Press 'Q' to quit\n");

printf("Please choose your next action:");

input=getche();

fflush(stdin);

if(input=='R' || input=='r')

{

system("cls");

row\_box+=0;/\*row\_box and col\_box are used to track current position and thus initialized to 0 at beginning.If 'R' is pressed,col\_box plus 1.\*/

col\_box+=1;

if(col\_box>7)

{

col\_box-=1;

}

else

{

row\_sel[++temp]=row\_box; /\*row\_box value is appended to row\_sel array at designated position using ++temp,so if temp is 1,row\_box value is saved into index 2 of row\_sel\*/

col\_sel[temp]=col\_box;

score+=matrix[row\_box][col\_box]; /\*Used to keep track of score\*/

if(row\_box==14 && col\_box==7) /\*Terminate game if position in grid is at point(14,7)\*/

{

for(row=0;row<15;row++)

{

for(col=0;col<8;col++)

{

foundMatch=false;

for(a=0;!foundMatch && a<sizeof(row\_sel)/(sizeof(row\_sel[0]));a++)

{

foundMatch=row==row\_sel[a] && col==col\_sel[a];

}

if(foundMatch)

{

if(col==7)

{

printf("-|%d|-",matrix[row][col]);

}

else

{

printf("-|%d|",matrix[row][col]);

}

}

else

{

if(row==1||row==3||row==5||row==7||row==9||row==11||row==13)

{

printf(" | ");

}

else

{

if(col==7)

{

printf("- %d -",matrix[row][col]);

}

else

{

printf("- %d ",matrix[row][col]);

}

}

}

}

printf("\n");

}

loop1=1;

}

}

}

if(input=='D'|| input=='d')

{

system("cls");

row\_box+=2;

col\_box+=0;

if(row\_box>14)

{

row\_box-=2;

}

else

{

row\_sel[++temp]=row\_box;

col\_sel[temp]=col\_box;

score+=matrix[row\_box][col\_box];

if(row\_box==14 && col\_box==7)

{

for(row=0;row<15;row++)

{

for(col=0;col<8;col++)

{

foundMatch=false;

for(a=0;!foundMatch && a<sizeof(row\_sel)/(sizeof(row\_sel[0]));a++)

{

foundMatch=row==row\_sel[a] && col==col\_sel[a];

}

if(foundMatch)

{

if(col==7)

{

printf("-|%d|-",matrix[row][col]);

}

else

{

printf("-|%d|",matrix[row][col]);

}

}

else

{

if(row==1||row==3||row==5||row==7||row==9||row==11||row==13)

{

printf(" | ");

}

else

{

if(col==7)

{

printf("- %d -",matrix[row][col]);

}

else

{

printf("- %d ",matrix[row][col]);

}

}

}

}

printf("\n");

}

loop1=1;

}

}

}

if(input=='Q' || input=='q')

{

loop1=1;

}

}

return score;

}

void endScreen(int score,fname \*end)

{

printf("Congratulation! You have completed your tour!\n\n");

printf("Total attractions you have visited: %d\n\n",score);

printf("Thanks for playing!! See you again...");

writeFile((\*end).name,score);/\*Writing player name and score to text file\*/

\_getch();

}

int main()

{

int a;

fname init;

startScreen(&init);

a=grid();

endScreen(a,&init);

return 0;

}