Stephanie Wyckoff  
Account: swyckoff1  
CSc 3320  
Program #3  
Due date: 10/20/16

Script started, file is hmwk3\_swyckoff1.log

[GSUAD\swyckoff1@snowball ~]$ cat hmwk3\_swyckoff1.c

/\*

\* hmwk3\_swyckoff1.c

\*

\* Stephanie Wyckoff

\* Account: swyckoff1

\* CSc 3320

\* Program #3

\* Due date: 10/20/16

\*

\* Discription: This program randomly generates paths through a 12x12 matrix using the letters of the alphabet from A-Z, a-z. A random number is chosen and modulo by 4 to provide the numerical value of the directions 0 = up, 1 = right, 2 = down, 3 = left. If the path is blocked by a letter already in the direction chosen, the program will find another direction for it to go in. If all paths are blocked, the program ends.

\*

\* Input: There is no input for this program.

\*

\* Output: The program generates the final path of letters through the matrix.

\*

\*/

#include <stdio.h>

#include <time.h>

#include <stdlib.h>

#define ROWS 12

#define COLS 12

main(int argc, char \*argv[]) {

int random ;

int dir;

char grid[ROWS][COLS];

char path[52] = {'A','B','C','D','E','F','G','H','I','J','K','L','M','N','O' ,'P','Q','R','S','T','U','V','W','X','Y','Z','a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t','u','v','w','x','y','z'};

srand(time(NULL));

int row=0;

int col=0;

int letter = 0;

//creates initial array of 12x12 and fills each slot with '.'.

for(row=0; row < ROWS; row++){

for(col=0; col < COLS; col++){

grid[row][col] = '.';

}

}

row = 0; col = 0;

grid[row][col] = path[letter]; // sets starting position of path at (0,0) and assigns it the value of 'A'

letter++;

while(letter < 52){

dir = rand() % 4;//generates random number and then modulo by 4 to produce 0,1,2 or 3

switch(dir){

case 0://checks one space up for open space

if(grid[row-1][col] == '.' && row != 0){

grid[row-1][col] = path[letter];//assigns letter to space

row--;//updates grid positon

}

if(grid[row-1][col] != '.' && grid[row][col+1] != '.' && grid[row+1][col] != '.' && grid[row][col-1] != '.'){

letter = 52;

break;

}

case 1://checks one space right for open space

if(grid[row][col+1] == '.' && col != COLS-1){

grid[row][col+1] = path[letter];//assigns letter to space

letter++; col++;//updates grid positon

}

if(grid[row-1][col] != '.' && grid[row][col+1] != '.' && grid[row+1][col] != '.' && grid[row][col-1] != '.'){

letter = 52;

break;

}

case 2://checks one space down for open space

if(grid[row+1][col] == '.' && row != ROWS-1){

grid[row+1][col] = path[letter];//assigns letter to space

letter++; row++;//updates grid position

}

if(grid[row-1][col] != '.' && grid[row][col+1] != '.' && grid[row+1][col] != '.' && grid[row][col-1] != '.'){

letter = 52;

break;

}

case 3://checks one space left for open space

if(grid[row][col-1] == '.' && col != 0){

grid[row][col-1] = path[letter];//assigns letter to space

letter++; col--;//updates grid position

}

if(grid[row-1][col] != '.' && grid[row][col+1] != '.' && grid[row+1][col] != '.' && grid[row][col-1] != '.'){

letter = 52;

break;

}

}

}

//prints the final grid with all letters of the path

for(row = 0; row < ROWS; row++){

for(col = 0; col < COLS; col++){

printf("%2c", grid[row][col]);

}printf("\n");

}

}

GSUAD\swyckoff1@snowball ~]$ gcc hmwk3\_swyckoff1.c -o hmwk3\_swyckoff1

GSUAD\swyckoff1@snowball ~]$ ./hmwk3\_swyckoff1

A B . . . . . . . . . .

D C . . . . . . . . . .

E F . . . . . . . . . .

H G . . . . . . . . . .

I J . . . . . . . . . .

L K . . . . . . . . . .

M N . . . . . . . . . .

P O . . . . . . . . . .

Q R . . . . . . . . . .

T S . . . . . . . . . .

U X X a a . . f f . j j

V W Y Z b c d e g h i k

[GSUAD\swyckoff1@snowball ~]$ ./hmwk3\_swyckoff1

A B . . . . . . . . . .

D C . . . . . . . . . .

E F . . . . . . . . . .

H G . . . . . . . . . .

I . . . . . . . . . . .

J K . . . . . . . . . .

M L . . . . . . . . . .

N O . . . . . . o o r r

Q P . . . . . n n p q s

R . . . . . . m l k j j

S V V . Z Z . d d g g j

T U W X Y a b c e f h i

[GSUAD\swyckoff1@snowball ~]$ mail -s "Homework 3" sakkineni1@student.gsu.edu < hmwk3\_swyckoff1.c

[GSUAD\swyckoff1@snowball ~]$ exit

exit

Script done, file is hmwk3\_swyckoff1.log

[GSUAD\swyckoff1@snowball ~]$