Bingo programme

Procedural programming project

Conor Raftery

G00274094

This word file includes:

* Code for running programme
* Readme

# Readme:

1. Copy and paste below code into visual studios.
2. Run the programme.
3. Select option 1 to start a new game.
4. Enter number of players. (Programme will only allow 1 player).
5. The programme will display a randomly generated bingo card (without the blank spaces, numbers may be duplicated). The bingo card is a 3x9 array. Each column has correctly generated numbers. E.g. Column 1 has numbers ranged 1-9, column 2 has numbers ranged 10-19. Etc.
6. Select option 1 to draw a number.
7. The programme will draw a random number ranged between 1-90. This number will then be checked against the players bingo card and will notify the user if the number is on (or not on) the players card.
8. Keep drawing new numbers.
9. If you wish to see all the numbers that have been drawn, select option 3.
10. This will output to the user all the numbers that have been drawn.
11. You can continue drawing numbers, or, you can save the game.
12. To save the game, select option 2.
13. After selecting this option, you will be notified that the game has been saved in a file named ‘bingo.txt’. You will also be shown how many numbers have been drawn in that game. Remember this number as you will need it if you wish to load that game at a later date.
14. If you wish, you can continue drawing numbers / saving the game / outputting drawn numbers. You can exit the programme by selecting option 4.
15. Exit the programme and run it again.
16. Select option 2 (load game).
17. You will be prompted to enter to amount of numbers that have been called in the game you wish to load. (Remember this number from earlier, step 13).
18. Enter number.
19. You will be shown the bingo card which has been saved, along with all the numbers that have been drawn.
20. You can continue the game as normal.
21. Option 4 to exit the programme.

I decided to put a random number into each index of the array manually, as I found it difficult to input a certain range of numbers into each column whilst using an inner for loop.

When comparing the numbers drawn to the card array, I only looped through the 1D array as I found it difficult to compare a 1D array with a 2D array. That is why it is hardcoded to compare every index of the 2D array.

I was going to use a 3D array to allow the user to select a certain number of players but I only figured this out when I had a lot of code written, which would be too much of a headache to change around.

I decided to keep the entire programme in 2 separate while loops, as it was easier to keep counters and such. I was going to use methods/functions but I found that my initial plan was better/easier.

# Code:

/\*

Title: Bingo Project

Name: Conor Raftery

Student No.: G00274094

Aim: Create a bingo programme using the C language, which creates a random player card, which calls random numbers and compares

them to the players card. The programme should be able to save a game, and load a previous game.

It should also be able to output the current games status.

\*/

#include<stdio.h>

#include<conio.h>

#include<stdlib.h>

#include <time.h>

main(void) {

//Gives me different random numbers every time program is ran

srand(time(NULL));

//Declare Variables

int bingo[3][9];

int bingoNumber[99];

int row;

int col;

int num = 0;

int noOfPlayers = 0;

int i = 0;

int j = 0;

int starterOption = 0;

int option = 0;

int counter = 0;

int fileBingo;

int calledNumbers = 0;

//Declares file pointer

FILE\* filep;

//---------------------------------------------------------------//

//Prompt for option

printf("Please Select Option:\n 1) Start New Game\n "

"2) Load Previous Game\n");

scanf("%d", &option);

//---------------------------------------------------------------//

//Option 1 - Start new game

if (option == 1) {

//Request number of players - only works with 1 player

printf("\nPlease Enter Number of Players: ");

scanf("%d", &noOfPlayers);

for (i = 0; i < noOfPlayers; i++) {

//Code for extra players

//Should have created 3D array - bingo[i][j][k];

}

printf("\nPlayer %d Bingo Card: \n", i);

//Place random number for each index

bingo[0][0] = (rand() % 10) + 1;//Top left corner

bingo[0][1] = (rand() % 10) + 11;

bingo[0][2] = (rand() % 10) + 21;

bingo[0][3] = (rand() % 10) + 31;

bingo[0][4] = (rand() % 10) + 41;

bingo[0][5] = (rand() % 10) + 51;

bingo[0][6] = (rand() % 10) + 61;

bingo[0][7] = (rand() % 10) + 71;

bingo[0][8] = (rand() % 10) + 81;//Bottom left corner

bingo[1][0] = (rand() % 10) + 1;

bingo[1][1] = (rand() % 10) + 11;

bingo[1][2] = (rand() % 10) + 21;

bingo[1][3] = (rand() % 10) + 31;

bingo[1][4] = (rand() % 10) + 41;

bingo[1][5] = (rand() % 10) + 51;

bingo[1][6] = (rand() % 10) + 61;

bingo[1][7] = (rand() % 10) + 71;

bingo[1][8] = (rand() % 10) + 81;

bingo[2][0] = (rand() % 10) + 1;//Top right corner

bingo[2][1] = (rand() % 10) + 11;

bingo[2][2] = (rand() % 10) + 21;

bingo[2][3] = (rand() % 10) + 31;

bingo[2][4] = (rand() % 10) + 41;

bingo[2][5] = (rand() % 10) + 51;

bingo[2][6] = (rand() % 10) + 61;

bingo[2][7] = (rand() % 10) + 71;

bingo[2][8] = (rand() % 10) + 81;//Bottom right corner

//Output bingoCard array

for (row = 0; row < 3; row++)

{

for (col = 0; col < 9; col++)

{

printf("%d\t", bingo[row][col]);

}

printf("\n");

}

//While loop for exiting programme

while (option != 4) {

//2nd option list

printf("\nPlease Select an option:\n "

"1) Draw Number\n"

"2) Save Game\n"

"3) Output current games status\n"

"4) Exit Program\n");

scanf("%d", &option);

//\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_//

//option 1 - draw number

if (option == 1) {

bingoNumber[j] = (rand() % 90) + 1; //Draws random number between 1-90

printf("\nBingo Number: %d \n", bingoNumber[j]); //Outputs number drawn

counter++; //Counter for ammount of drawn numbers

j++;//Counter for BingoNumber

i = 0;

//Loops through each completed iteration

while (i < counter) {

//Compares numbers drawn with numbers in card

if (bingoNumber[i] == bingo[0][0] || bingoNumber[i] == bingo[0][1] || bingoNumber[i] == bingo[0][2] || bingoNumber[i] == bingo[0][3] || bingoNumber[i] == bingo[0][4] || bingoNumber[i] == bingo[0][5] || bingoNumber[i] == bingo[0][6] || bingoNumber[i] == bingo[0][7] || bingoNumber[i] == bingo[0][8]

|| bingoNumber[i] == bingo[1][0] || bingoNumber[i] == bingo[1][1] || bingoNumber[i] == bingo[1][2] || bingoNumber[i] == bingo[1][3] || bingoNumber[i] == bingo[1][4] || bingoNumber[i] == bingo[1][5] || bingoNumber[i] == bingo[1][6] || bingoNumber[i] == bingo[1][7] || bingoNumber[i] == bingo[1][8]

|| bingoNumber[i] == bingo[2][0] || bingoNumber[i] == bingo[2][1] || bingoNumber[i] == bingo[2][2] || bingoNumber[i] == bingo[2][3] || bingoNumber[i] == bingo[2][4] || bingoNumber[i] == bingo[2][5] || bingoNumber[i] == bingo[2][6] || bingoNumber[i] == bingo[2][7] || bingoNumber[i] == bingo[2][8]) {

printf("\nBingo Number %d marked on card\n", bingoNumber[i]);// if number in card has been found, output to screen

}

else {

printf("\nNumber %d not marked on card\n", bingoNumber[i]);//if number has not been found, output to screen

}

i++;

}//while

}//option1

//\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_//

//Option2 - To save game

if (option == 2) {

printf("\nGame Saved! See 'Bingo.txt'\n");

//Outputs total of drawn numbers

printf("\nThe total of numbers drawn are %d,\n Please take note of this if you wish to load the game!\n", counter);

//Create/open file

filep = fopen("Bingo.txt", "w");

//Output bingoCard to file

for (row = 0; row < 3; row++)

{

for (col = 0; col < 9; col++)

{

fprintf(filep, "%d\t", bingo[row][col]);

}

fprintf(filep, "\n");

}

for (i = 0; i < counter; i++) {

fprintf(filep, "%d \n", bingoNumber[i]);

}

//close file

fclose(filep);

}//Option2

//\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_//

//Option3 - Outputs numbers that have been drawn

if (option == 3) {

printf("\nNumbers Drawn: \n");

//loops through every number which has been drawn and outputs

for (i = 0; i < counter; i++) {

printf("%d \n", bingoNumber[i]);

}

}//Option 3

}//end while

}//End starter option 1

//========================================================================================================//

else if (option == 2) {

//Prompt user for count for 1D array

printf("\nPlease enter the amount of numbers called in previous game: \n");

scanf("%d", &calledNumbers);

//opens file for reading

filep = fopen("Bingo.txt", "r");

//first if reads unil file is null via while loop

if (filep != NULL) {

while (!feof(filep)) {

//reads file data into variables

for (row = 0; row < 3; row++) {

for (col = 0; col < 9; col++) {

fileBingo = fscanf(filep, "%d", &bingo[row][col]);

}

}

for (i = 0; i < calledNumbers; i++) {

fileBingo = fscanf(filep, "%d", &bingoNumber[i]);

}

printf("\nGame Loaded!\n");

//when reaches end of file, outputs to screen

if (fileBingo > 0) {

printf("\nBingo Card: \n");

for (row = 0; row < 3; row++) {

for (col = 0; col < 9; col++) {

printf("%d\t", bingo[row][col]);

}

printf("\n");

}

for (i = 0; i < calledNumbers; i++) {

fileBingo = printf("%d\n", bingoNumber[i]);

}

}

}//while

//close file

fclose(filep);

}//first if

//if file is empty, print out file is empty to screen

else

{

printf("File is empty\n");

}

//close file

fclose(filep);

//----------------------------------------------------------------------//

while (option != 4) {

printf("\nPlease Select an option:\n "

"1) Draw Number\n"

"2) Save Game\n"

"3) Output current games status\n"

"4) Exit Program\n");

scanf("%d", &option);

//\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_//

//option 1 - draw number

if (option == 1) {

bingoNumber[j] = (rand() % 90) + 1; //Draws random number between 1-90

printf("\nBingo Number: %d \n", bingoNumber[j]); //Outputs number drawn

counter++; //Counter for ammount of drawn numbers

j++;//Counter for BingoNumber

i = 0;

//Loops through each completed iteration

while (i < counter) {

//Compares numbers drawn with numbers in card

if (bingoNumber[i] == bingo[0][0] || bingoNumber[i] == bingo[0][1] || bingoNumber[i] == bingo[0][2] || bingoNumber[i] == bingo[0][3] || bingoNumber[i] == bingo[0][4] || bingoNumber[i] == bingo[0][5] || bingoNumber[i] == bingo[0][6] || bingoNumber[i] == bingo[0][7] || bingoNumber[i] == bingo[0][8]

|| bingoNumber[i] == bingo[1][0] || bingoNumber[i] == bingo[1][1] || bingoNumber[i] == bingo[1][2] || bingoNumber[i] == bingo[1][3] || bingoNumber[i] == bingo[1][4] || bingoNumber[i] == bingo[1][5] || bingoNumber[i] == bingo[1][6] || bingoNumber[i] == bingo[1][7] || bingoNumber[i] == bingo[1][8]

|| bingoNumber[i] == bingo[2][0] || bingoNumber[i] == bingo[2][1] || bingoNumber[i] == bingo[2][2] || bingoNumber[i] == bingo[2][3] || bingoNumber[i] == bingo[2][4] || bingoNumber[i] == bingo[2][5] || bingoNumber[i] == bingo[2][6] || bingoNumber[i] == bingo[2][7] || bingoNumber[i] == bingo[2][8]) {

printf("\nBingo Number %d marked on card\n", bingoNumber[i]);// if number in card has been found, output to screen

}

else {

printf("\nNumber %d not marked on card\n", bingoNumber[i]);//if number has not been found, output to screen

}

i++;

}//while

}//option1

//\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_//

//Option2 - To save game

if (option == 2) {

printf("Game Saved! See 'Bingo.txt'");

//Outputs total of drawn numbers

printf("\nThe total of numbers drawn are %d,\n Please take note of this if you wish to load the game!\n", counter);

//Create/open file

filep = fopen("Bingo.txt", "w");

//Output bingoCard to file

for (row = 0; row < 3; row++)

{

for (col = 0; col < 9; col++)

{

fprintf(filep, "%d\t", bingo[row][col]);

}

fprintf(filep, "\n");

}

for (i = 0; i < counter; i++) {

fprintf(filep, "%d \n", bingoNumber[i]);

}

//close file

fclose(filep);

}//Option2

//\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_//

//Option3 - Outputs numbers that have been drawn

if (option == 3) {

printf("\nNumbers Drawn: \n");

//loops through every number which has been drawn and outputs

for (i = 0; i < counter; i++) {

printf("%d \n", bingoNumber[i]);

}

}//Option 3

}//First while

}//End Starter option 2

getch();

}//end main