

Anton Hall 1809131 - ACCA4017

by Anton Hall

FILE	12316_ANTON_HALL_ANTON_HALL_1809131_-_ACCA4017_173926_1520881354.DOCX (516.67K)		
TIME SUBMITTED	01-MAY-2019 09:21PM (UTC+0100)	WORD COUNT	2160
SUBMISSION ID	106019886	CHARACTER COUNT	10172

University of Wales, Trinity Saint David

Software Development – ACCA4017

Programs in C and Documentation (Part C)

Anton Hall

Contents

Introduction	1
1. Part C.....	1
1.1 Problem.....	1
1.2 Pseudocode.....	1
1.3 Annotated Code	2
1.4 Testing.....	6
1.4.1 Normal	6
1.4.2 Erroneous.....	13
Conclusion.....	14

Introduction

The purpose of this assignment is to demonstrate the development and testing of a C program that allows a user to play a version of a popular card matching game.

1. Part C

1.1 Problem

This program is intended to showcase all different elements of C that have been taught this semester as part of the software development module. The program needs to be a version of a popular card matching game. As this program is a game, the pairs must be in a different order each time the game is played, so random numbers will need to be generated in order to swap the positions of the pairs in the array. The player will need to be able to turn over two cards at a time. If these two cards are a match, then we will need to tell the player and illustrate to them that these are now a match and can't be picked again, decisions such as If and else can be used to do this. It is also a requirement to use functions within the program. As arrays cannot be passed to different functions either a global variable will need to be declared for the array (not best practise) or pointers will need to be used to pass and return the array to multiple functions. As the aim of the game is to get the most pairs with the least amount of turns, scores will need to be recorded throughout the game so the player can be told their result at the end, incorporating while loops will help here. We could also incorporate a current high score, but this will need to be stored in an external file.

1.2 Pseudocode

main

```
set play = x
set pairs = 1,1,2,2,3,3,4,4,5,5,6,6
print Welcome to the game, do you want to play?
Store play
```

```
while play is Y
    random (pairs)
    game (pairs)
    print Do you want to play again.
    Store play
```

Random (pairs)

```
set i, j, k, temp, high, low
for 0 – 100 (counting loop)
    set j = random number between low and high
    set k = random number between low and high
    set temp = pairs(j)
    set pairs(j) = pairs(k)
    set pairs(k) = temp
return pairs to main
```

game(pairs)

```
set i, matched Pairs = 0, card1, card2, turn Count = 0, remaining = 1,2,3,4,5,6,7,8,9,10,11,12,
used = -, -, -, -, -, -, -, -, -, -, -, -
```

```

while matched Pairs less than 6
    print Positions Remaining:
    print Positions used:
    print Pick a Card Position
    store card1
    print Pick a Second Card Position
    store card2
    while card1 = card2
        print You entered the same card as number one, please enter your Second
        card Position
        store card2
    card1 = card1 - 1
    card2 = card2 - 1
    if pairs(card1) = pairs(card2) & used(card1) != M & used(card2) != M
        print Its a match, you matched the 's
        set used(card1) = M
        set used(card2) = M
        set matched Pairs = matched Pairs + 1

    else if pairs(card1) = pairs(card2) & used(card1) = M & used(card2) = M

        Print You already matched these, please try again

    else

        print Try Again! you picked a ? & ?

    set turn Count = turn Count + 1

    if matched Pairs = 6

        scoreboard(turn Count)

scoreboard(turn Count)
    set current high, file

    file = open High Score
    read file for current high

    if turn Count < current high
        file = open High Score
        file = write turn Count
        print Well done you got a new high score of: ? turns

    else

        print You didn't get a high score, better luck next time

```

1.3 Annotated Code

```

#include <stdio.h> //declare libraries
#include <stdlib.h>

```

```

#include <time.h>

int random(int*); //prototype
void game(int*); //prototype
void scoreboard(int); //prototype

void main() { //main function
    char play = 'x'; //declare play to be x
    int pairs[12] = { 1,1,2,2,3,3,4,4,5,5,6,6 }; //declare the pairs variable
    srand(time(NULL)); //seed random with current time
    printf("****Welcome to the card matching game*** \nDo you want to play?
(Y/N):"); //Welcome player to the game and ask them if they want to play
    scanf(" %c", &play); //store players answer to previous question

    while (play == 'y' || play == 'Y') { //start a while loop to run while the
player wants to play
        random(pairs); //use the random function to randomise the order of the
pairs array - pass the pairs variable to the function
        game(pairs); //use the game function to allow the player to play the game
- pass the pairs variable to the function
        printf("\nDo you want to play again? (Y/N):"); // ask the player if they
want to play again
        scanf(" %c", &play); // store the users answer in the play variable
which will restart the current while loop if the player wants to play again
    }
}

int random(int *pairs) { //random function - set as integer as pairs will be passed
back to main, and set pairs variable as a pointer
    int i, j, k, temp, high = 11, low = 0; //declare variables
    for (i = 0; i < 100; i++) { //start a counting loop using the variable i to run
100 times
        j = (rand() % high - low + 1) + low; //get a random number between the
high and low values and store in variable j
        k = (rand() % high - low + 1) + low; //get a random number between the
high and low values and store in variable k
        temp = pairs[j]; //store the number held in the array at random index
number j in the temp variable
        pairs[j] = pairs[k]; //swap the value held in the array at index number
j with the value held at index number k
        pairs[k] = temp; //swap the value held previously in index j and now
stored in temp with value stored in index k
    }
    return *pairs; //return pairs as a pointer
}

void game(int *pairs) { //game function -set as void as nothing will be passed back to
main, and set pairs variable as a pointer

int i, matchedPairs = 0, card1, card2, turnCount = 0, remaining[12] = {
1,2,3,4,5,6,7,8,9,10,11,12 }; //declare variables
char used[12] = { '-', '-', '-', '-', '-', '-', '-', '-', '-', '-', '-', '-' }; //declare
variables
/*for testing..
for (i = 0; i < 12; i++) {
    printf("%d: %d", (i+1), *(pairs + i));
}
for testing..*/
    while (matchedPairs < 6) //start a while loop to run until the player has
matched all 6 pairs

```

```

{
    printf("\n\nPositions Remaining:\n"); //print the available positions
    available to the player
    printf("%d: %c      %d: %c %d: %c %d: %c      %d: %c      %d: %c\n",
        remaining[0], used[0], remaining[1], used[1], remaining[2], used[2],
        remaining[3], used[3], remaining[4], used[4], remaining[5], used[5]);
    //^ print the remaining and used cards
    printf("\n%d: %c      %d: %c %d: %c %d: %c %d: %c %d: %c ", remaining[6],
        used[6], remaining[7], used[7], remaining[8], used[8], remaining[9], used[9],
        remaining[10], used[10], remaining[11], used[11]);
    //^ print the remaining and used cards
    printf("\nPick a Card Position(1 - 12):\n"); //ask the play to pick the
    first card position between 1 and 12
    scanf("%d", &card1); //store the players choice in card1
    while (card1 < 1 || card1 > 2) { //check whether the integer entered is
        valid
        printf("invalid input please try again:"); //tell the user it is
        invalid
        scanf("%d", &card1); //store the players choice again
    }
    printf("Pick a Second Card Position(1 - 12):\n"); //ask the play to pick
    the second card position between 1 and 12
    scanf("%d", &card2); //store the players choice in card2
    while (card2 < 1 || card2 > 12){ //check whether the integer entered is
        valid
        printf("invalid input please try again:"); //tell the user it is
        invalid
        scanf("%d", &card2); //store the players choice again
    }
    card1 = card1 - 1; //subtract one from each of the players card choices
    as arrays start from 0
    card2 = card2 - 1;
    if(card1 == card2) //start a while loop that checks the player isnt
    picking the same card twice
    {
        printf("You entered the same card as number one, please enter
        your Second card Position(1 - 12):\n"); //tell the player they picked the same card
        twice and ask them for card2 again
        scanf("%d", &card2); //store the players choice in card2
    }
    else if (*(pairs + card1) == *(pairs + card2) && used[card1] != 'M' &&
    used[card2] != 'M') // start an if statement to check whether the player has matched a
    pair and that its not been matched already
    {
        printf("\nIts a match, you matched the %d's", *(pairs + card1));
        // tell the player they have matched and pair and what pair it was
        used[card1] = 'M'; //change the - to and M in the used array to
        show the player this card has been used
        used[card2] = 'M'; //change the - to and M in the used array to
        show the player this card has been used
        matchedPairs = matchedPairs + 1; //add 1 to the amount of matched
        pairs the player has made
    }
    else if (*(pairs + card1) == *(pairs + card2) && used[card1] == 'M' &&
    used[card2] == 'M') //next argument in the if statement that lets the player know they
    have already matched the cards they have picked
    {
        printf("\nYou already matched these, please try again"); //tell
        the player they have already matched these cards
    }
    else
    {

```

```

        printf("\nTry Again! you picked a %d & %d", *(pairs + card1),
*(pairs + card2)); //last argument for the if statement that tells the player they
have not matched any cards
    }
    turnCount = turnCount + 1; //add 1 the the amount of turns the player
has had in this game
    if (matchedPairs == 6) //start an if statement that will run the
scoreboard function once the player has made six matches
    {
        scoreboard(turnCount); //run the scorebaord function passing the
turncount variable to it
    }
}

void scoreboard(int turnCount) { //scoreboard function

    int currenthigh; //declare variables

    FILE *fp; //create pointer to communicate with a file

    fp = fopen("HighScore.txt", "r"); //open the HighScore.txt and read the current
high score
    fscanf(fp, "%d", &currenthigh); //store the high score in the currenthigh
variable
    fclose(fp); //close the file

    if (turnCount < currenthigh) { //start an if statement to check whether the
player has a new high score
        fp = fopen("HighScore.txt", "w"); //if a player does have a high score
open the HighScore.txt in write mode
        fprintf(fp, "%d", turnCount); //print the new highscore to the file
        fclose(fp); //close the file
        printf("\n ***Well done you got a new high score of: %d turns!***",
turnCount); //tell the player they got a new highscore and tell them what it is
    }
    else{ //last argument for the if - if the players turncount is greater than the
current highscore then

        printf("\nYou didnt get a high score, better luck next time..."); //tell
the player they didnt get a highscore this time
    }

}

    fscanf(fp, "%d", &currenthigh); //store the high score in the currenthigh
variable
    fclose(fp); //close the file

    if (turnCount < currenthigh) { //start an if statement to check whether the
player has a new high score
        fp = fopen("HighScore.txt", "w"); //if a player does have a high score
open the HighScore.txt in write mode
        fprintf(fp, "%d", turnCount); //print the new highscore to the file
        fclose(fp); //close the file
        printf("\n ***Well done you got a new high score of: %d turns!***",
turnCount); //tell the player they got a new highscore and tell them what it is
    }
    else{ //last argument for the if - if the players turncount is greater than the
current highscore then

```

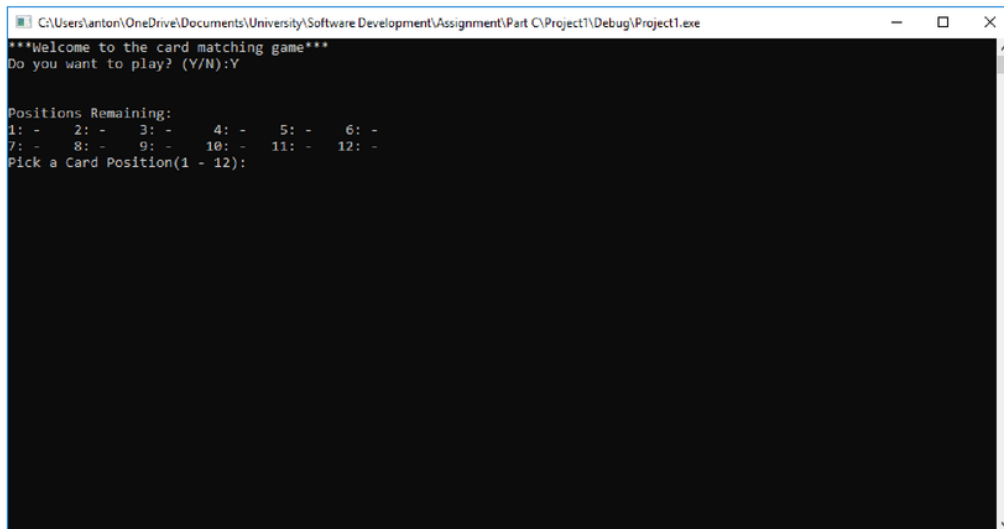
```
        printf("\nYou didnt get a high score, better luck next time..."); //tell  
the player they didnt get a highscore this time  
    }
```

1.4 Testing

Please note that the random pairs have been printed to the program in order to aid some parts of testing. These will be removed for normal game play.

1.4.1 Normal

If the user decides to play the game:

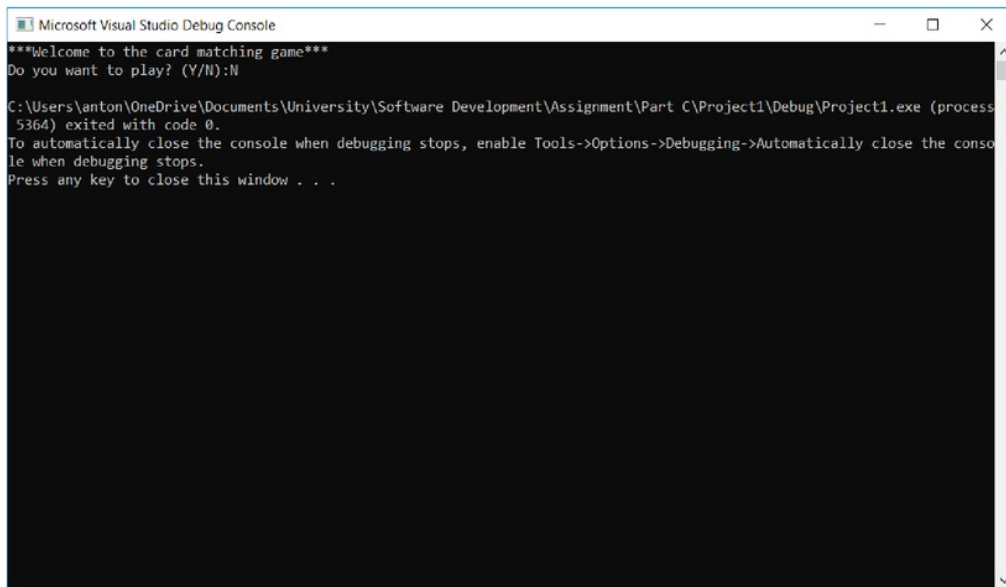


The screenshot shows a Windows command prompt window titled "C:\Users\anton\OneDrive\Documents\University\Software Development\Assignment\Part C\Project1\Debug\Project1.exe". The window contains the following text:

```
***Welcome to the card matching game***  
Do you want to play? (Y/N):Y  
  
Positions Remaining:  
1: - 2: - 3: - 4: - 5: - 6: -  
7: - 8: - 9: - 10: - 11: - 12: -  
Pick a Card Position(1 - 12):
```

Result: Program runs as expected.

If the user decides not to play the game:

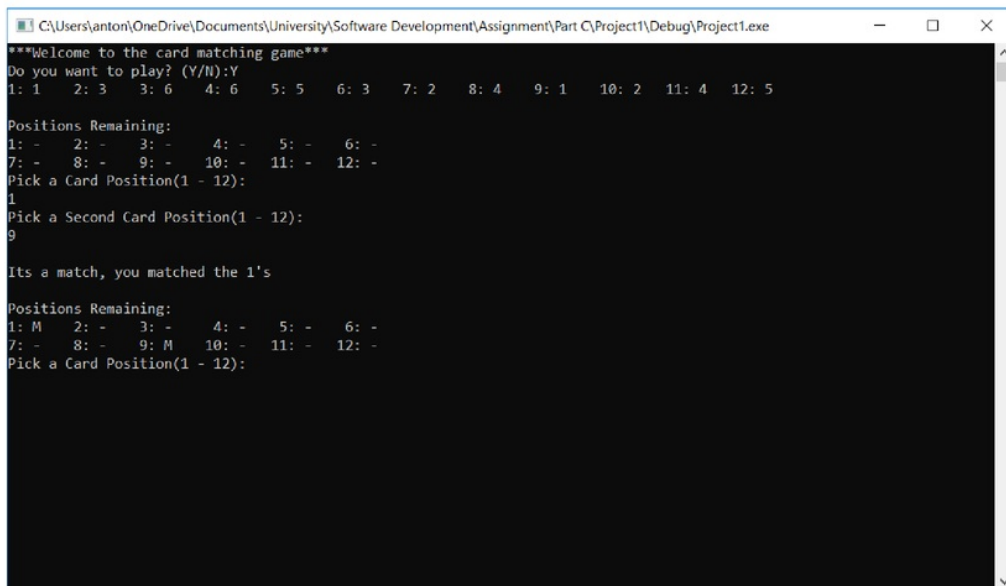


```
Microsoft Visual Studio Debug Console
***Welcome to the card matching game***
Do you want to play? (Y/N):N

C:\Users\anton\OneDrive\Documents\University\Software Development\Assignment\Part C\Project1\Debug\Project1.exe (process 5364) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```

Result: Program runs as expected.

If the user gets a match:



```
C:\Users\anton\OneDrive\Documents\University\Software Development\Assignment\Part C\Project1\Debug\Project1.exe
***Welcome to the card matching game***
Do you want to play? (Y/N):Y
1: 1  2: 3  3: 6  4: 6  5: 5  6: 3  7: 2  8: 4  9: 1  10: 2  11: 4  12: 5

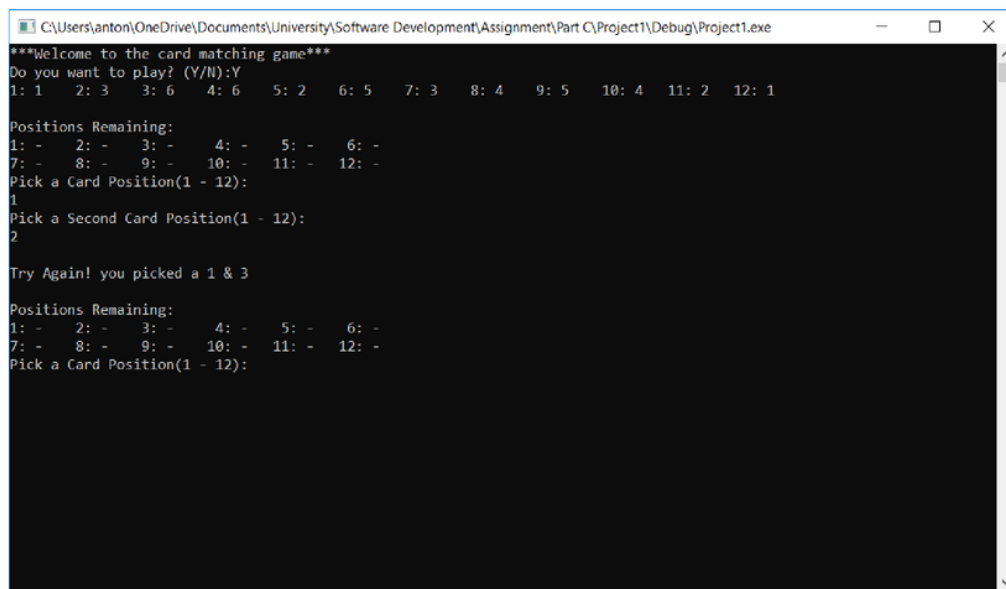
Positions Remaining:
1: -  2: -  3: -  4: -  5: -  6: -
7: -  8: -  9: -  10: -  11: -  12: -
Pick a Card Position(1 - 12):
1
Pick a Second Card Position(1 - 12):
9

Its a match, you matched the 1's

Positions Remaining:
1: M  2: -  3: -  4: -  5: -  6: -
7: -  8: -  9: M  10: -  11: -  12: -
Pick a Card Position(1 - 12):
```

Result: Program runs as expected.

If the user doesn't get a match:



```
C:\Users\anton\OneDrive\Documents\University\Software Development\Assignment\Part C\Project1\Debug\Project1.exe
***Welcome to the card matching game***
Do you want to play? (Y/N):Y
1: 1   2: 3   3: 6   4: 6   5: 2   6: 5   7: 3   8: 4   9: 5   10: 4   11: 2   12: 1

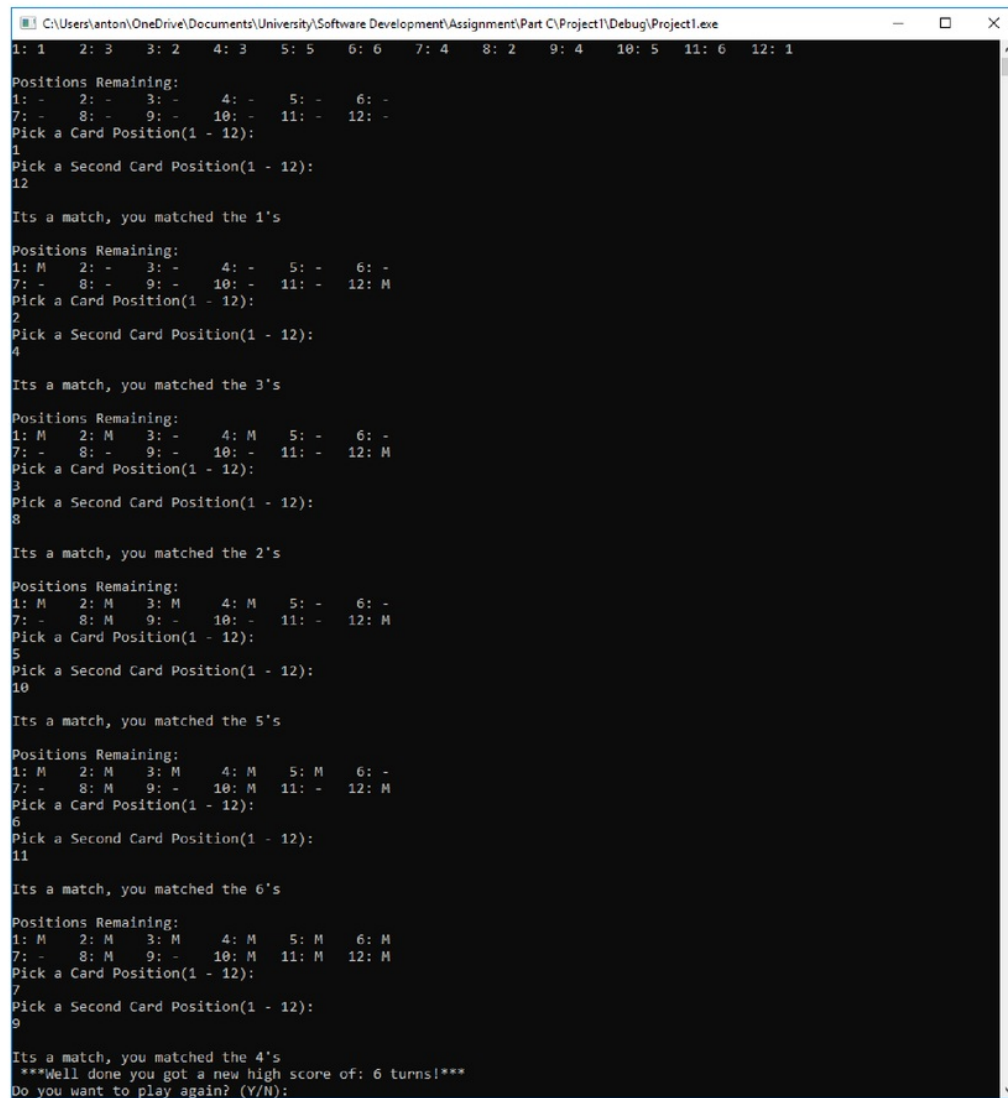
Positions Remaining:
1: -   2: -   3: -   4: -   5: -   6: -
7: -   8: -   9: -   10: -   11: -   12: -
Pick a Card Position(1 - 12):
1
Pick a Second Card Position(1 - 12):
2

Try Again! you picked a 1 & 3

Positions Remaining:
1: -   2: -   3: -   4: -   5: -   6: -
7: -   8: -   9: -   10: -   11: -   12: -
Pick a Card Position(1 - 12):
```

Result: Program runs as expected.

If the user gets a high score:



```
C:\Users\anton\OneDrive\Documents\University\Software Development\Assignment\Part C\Project\Debug\Project1.exe
1: 1 2: 3 3: 2 4: 3 5: 5 6: 6 7: 4 8: 2 9: 4 10: 5 11: 6 12: 1
Positions Remaining:
1: - 2: - 3: - 4: - 5: - 6: -
7: - 8: - 9: - 10: - 11: - 12: -
Pick a Card Position(1 - 12):
1
Pick a Second Card Position(1 - 12):
12
Its a match, you matched the 1's
Positions Remaining:
1: M 2: - 3: - 4: - 5: - 6: -
7: - 8: - 9: - 10: - 11: - 12: M
Pick a Card Position(1 - 12):
2
Pick a Second Card Position(1 - 12):
4
Its a match, you matched the 3's
Positions Remaining:
1: M 2: M 3: - 4: M 5: - 6: -
7: - 8: - 9: - 10: - 11: - 12: M
Pick a Card Position(1 - 12):
3
Pick a Second Card Position(1 - 12):
8
Its a match, you matched the 2's
Positions Remaining:
1: M 2: M 3: M 4: M 5: - 6: -
7: - 8: M 9: - 10: - 11: - 12: M
Pick a Card Position(1 - 12):
5
Pick a Second Card Position(1 - 12):
10
Its a match, you matched the 5's
Positions Remaining:
1: M 2: M 3: M 4: M 5: M 6: -
7: - 8: M 9: - 10: M 11: - 12: M
Pick a Card Position(1 - 12):
6
Pick a Second Card Position(1 - 12):
11
Its a match, you matched the 6's
Positions Remaining:
1: M 2: M 3: M 4: M 5: M 6: M
7: - 8: M 9: - 10: M 11: M 12: M
Pick a Card Position(1 - 12):
7
Pick a Second Card Position(1 - 12):
9
Its a match, you matched the 4's
***Well done you got a new high score of: 6 turns!***
Do you want to play again? (Y/N):
```

Result: Program runs as expected.

If the player doesn't get a high score:

```
C:\Users\anton\OneDrive\Documents\University\Software Development\Assignment\Part C\Project\Debug\Project1.exe
1: - 2: - 3: - 4: - 5: - 6: -
7: - 8: - 9: - 10: - 11: - 12: -
Pick a Card Position(1 - 12):
1
Pick a Second Card Position(1 - 12):
7
Its a match, you matched the 1's
Positions Remaining:
1: M 2: - 3: - 4: - 5: - 6: -
7: M 8: - 9: - 10: - 11: - 12: -
Pick a Card Position(1 - 12):
2
Pick a Second Card Position(1 - 12):
10
Its a match, you matched the 6's
Positions Remaining:
1: M 2: M 3: - 4: - 5: - 6: -
7: M 8: - 9: - 10: M 11: - 12: -
Pick a Card Position(1 - 12):
3
Pick a Second Card Position(1 - 12):
8
Its a match, you matched the 5's
Positions Remaining:
1: M 2: M 3: M 4: - 5: - 6: -
7: M 8: M 9: - 10: M 11: - 12: -
Pick a Card Position(1 - 12):
4
Pick a Second Card Position(1 - 12):
6
Its a match, you matched the 4's
Positions Remaining:
1: M 2: M 3: M 4: M 5: - 6: M
7: M 8: M 9: - 10: M 11: - 12: -
Pick a Card Position(1 - 12):
5
Pick a Second Card Position(1 - 12):
12
Its a match, you matched the 2's
Positions Remaining:
1: M 2: M 3: M 4: M 5: M 6: M
7: M 8: M 9: - 10: M 11: - 12: M
Pick a Card Position(1 - 12):
9
Pick a Second Card Position(1 - 12):
11
Its a match, you matched the 3's
You didnt get a high score, better luck next time...
Do you want to play again? (Y/N):
```

Result: Program runs as expected.

If the player wants to play again:

```
C:\Users\anton\OneDrive\Documents\University\Software Development\Assignment\Part C\Project1\Debug\Project1.exe
Pick a Second Card Position(1 - 12):
8
Its a match, you matched the 2's
Positions Remaining:
1: M 2: M 3: M 4: M 5: - 6: -
7: - 8: M 9: - 10: - 11: - 12: M
Pick a Card Position(1 - 12):
5
Pick a Second Card Position(1 - 12):
10
Its a match, you matched the 5's
Positions Remaining:
1: M 2: M 3: M 4: M 5: M 6: -
7: - 8: M 9: - 10: M 11: - 12: M
Pick a Card Position(1 - 12):
6
Pick a Second Card Position(1 - 12):
11
Its a match, you matched the 6's
Positions Remaining:
1: M 2: M 3: M 4: M 5: M 6: M
7: - 8: M 9: - 10: M 11: M 12: M
Pick a Card Position(1 - 12):
7
Pick a Second Card Position(1 - 12):
9
Its a match, you matched the 4's
***Well done you got a new high score of: 6 turns!***
Do you want to play again? (Y/N):Y
1: 1 2: 3 3: 6 4: 4 5: 1 6: 4 7: 2 8: 5 9: 6 10: 5 11: 3 12: 2
Positions Remaining:
1: - 2: - 3: - 4: - 5: - 6: -
7: - 8: - 9: - 10: - 11: - 12: -
Pick a Card Position(1 - 12):
1
Pick a Second Card Position(1 - 12):
5
Its a match, you matched the 1's
Positions Remaining:
1: M 2: - 3: - 4: - 5: M 6: -
7: - 8: - 9: - 10: - 11: - 12: -
Pick a Card Position(1 - 12):
2
Pick a Second Card Position(1 - 12):
4
Try Again! you picked a 3 & 4
Positions Remaining:
1: M 2: - 3: - 4: - 5: M 6: -
7: - 8: - 9: - 10: - 11: - 12: -
Pick a Card Position(1 - 12):
```

Result: Program runs as expected.

Player chooses the same card twice:

```
C:\Users\anton\OneDrive\Documents\University\Software Development\Assignment\Part C\Project1\Debug\Project1.exe
***Welcome to the card matching game***
Do you want to play? (Y/N):Y
1: 1  2: 3  3: 4  4: 2  5: 6  6: 3  7: 4  8: 6  9: 2  10: 5  11: 5  12: 1

Positions Remaining:
1: -  2: -  3: -  4: -  5: -  6: -
7: -  8: -  9: -  10: -  11: -  12: -
Pick a Card Position(1 - 12):
1
Pick a Second Card Position(1 - 12):
1
You entered the same card as number one, please enter your Second card Position(1 - 12):
2

Try Again! you picked a 1 & 4

Positions Remaining:
1: -  2: -  3: -  4: -  5: -  6: -
7: -  8: -  9: -  10: -  11: -  12: -
Pick a Card Position(1 - 12):
```

Result: Program runs as expected.

Player chooses a set of cards that has already been matched:

```
Select C:\Users\anton\OneDrive\Documents\University\Software Development\Assignment\Part C\Project1\Debug\Project1.exe
***Welcome to the card matching game***
Do you want to play? (Y/N):Y
1: 1  2: 3  3: 5  4: 3  5: 2  6: 5  7: 2  8: 1  9: 6  10: 4  11: 4  12: 6

Positions Remaining:
1: -  2: -  3: -  4: -  5: -  6: -
7: -  8: -  9: -  10: -  11: -  12: -
Pick a Card Position(1 - 12):
1
Pick a Second Card Position(1 - 12):
8

Its a match, you matched the 1's

Positions Remaining:
1: M  2: -  3: -  4: -  5: -  6: -
7: -  8: M  9: -  10: -  11: -  12: -
Pick a Card Position(1 - 12):
1
Pick a Second Card Position(1 - 12):
8

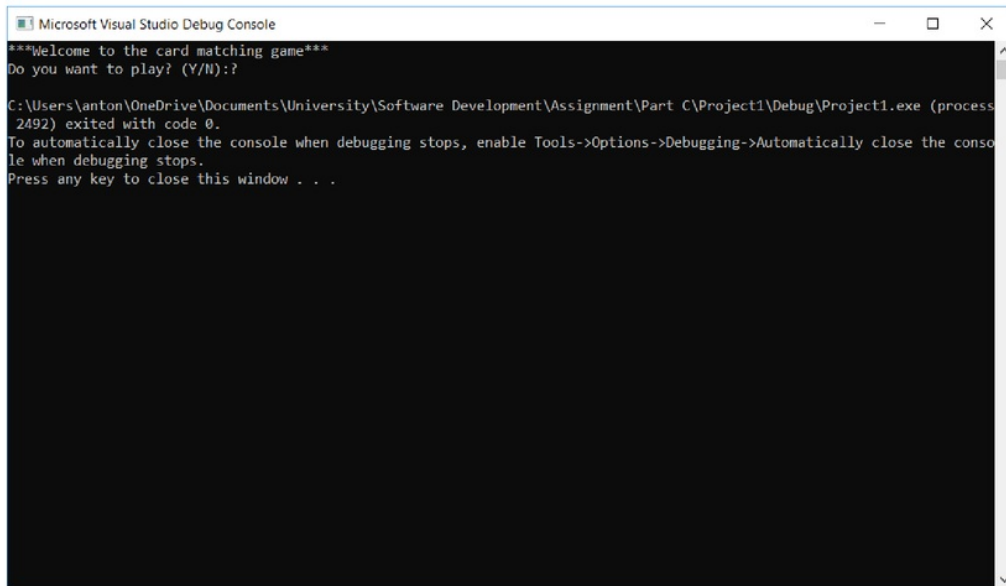
You already matched these, please try again

Positions Remaining:
1: M  2: -  3: -  4: -  5: -  6: -
7: -  8: M  9: -  10: -  11: -  12: -
Pick a Card Position(1 - 12):
```

Result: Program runs as expected.

1.4.2 Erroneous

Player uses wrong values to play the game:

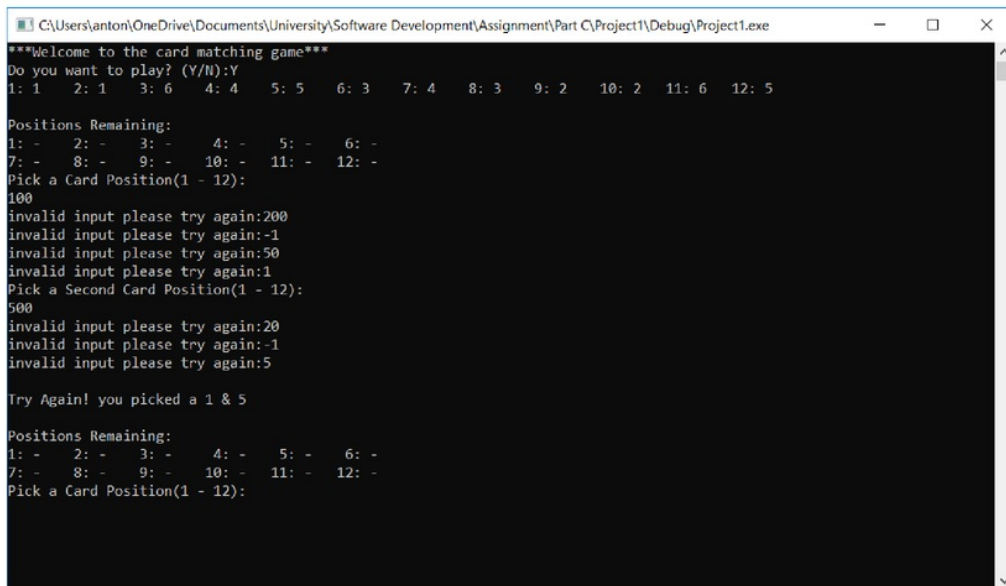


```
Microsoft Visual Studio Debug Console
***Welcome to the card matching game***
Do you want to play? (Y/N):?

C:\Users\anton\OneDrive\Documents\University\Software Development\Assignment\Part C\Project1\Debug\Project1.exe (process
2492) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console
when debugging stops.
Press any key to close this window . . .
```

Result: Program runs as expected.

Player chooses an invalid card number:



```
C:\Users\anton\OneDrive\Documents\University\Software Development\Assignment\Part C\Project1\Debug\Project1.exe
***Welcome to the card matching game***
Do you want to play? (Y/N):Y
1: 1  2: 1  3: 6  4: 4  5: 5  6: 3  7: 4  8: 3  9: 2  10: 2  11: 6  12: 5

Positions Remaining:
1: -  2: -  3: -  4: -  5: -  6: -
7: -  8: -  9: -  10: -  11: -  12: -
Pick a Card Position(1 - 12):
100
invalid input please try again:200
invalid input please try again:-1
invalid input please try again:50
invalid input please try again:1
Pick a Second Card Position(1 - 12):
500
invalid input please try again:20
invalid input please try again:-1
invalid input please try again:5

Try Again! you picked a 1 & 5

Positions Remaining:
1: -  2: -  3: -  4: -  5: -  6: -
7: -  8: -  9: -  10: -  11: -  12: -
Pick a Card Position(1 - 12):
```

Result: Program runs as expected.

Conclusion

The above report demonstrates the development of a program in c that allows a user to play a version of a well-known card matching game. It identifies the specific problems faced when completing the exercise and demonstrates the logical stages that are required for the program to run effectively. It shows the code that was written for the program along with annotations and testing that ensures the program performs as it is intended to do so.

FINAL GRADE

/100

GENERAL COMMENTS

Instructor

PART C

Coding/Annotation- Structure is a bit confused in my opinion - the game() function is really like a main and needed further breakdown. At times you are a bit inefficient eg where you print out the remaining positions, a loop would have been better here. I think there's a mistake on your validation for card 1 - it validates between 1 and 2 not 1 and 12. Overall a good attempt, could have been simplified in parts 15/20

Documentation - A logical breakdown is needed, you are very close to actual code. Testing is OK - make sure you test the boundaries of your if statements - for example 1 and 12 on the card input. This would have teased out the error above. 10/15

Total 25/35

Overall total for assignment: 46.5+25

71.5%

Test mark 90%

Overall mark for module 77%

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15

PAGE 16
