

C Programs and Documentation

by Nicholas Regan

FILE	12125_NICHOLAS_REGAN_C_PROGRAMS_AND_DOCUMENTATION_173926_1248738335.DOCX (212.47K)		
TIME SUBMITTED	02-MAY-2019 02:04 PM (UTC+0100)	WORD COUNT	557
SUBMISSION ID	106099703	CHARACTER COUNT	2556

Title: Programmes in C and Documentation
Part C: (Implementation-Function Based)
C Programme of a Card Game

Module: Software Development
ACCA4017

Course: HND Electronics and Electrical Engineering

University of Wales Trinity St David

By: Nick Regan
Student No: 1803871
Date: 2.5.19

Declaration

The work of this report completed under the supervision of Gaynor Thomas

I declare that this report is the product of my own work and is not being submitted for any other degree etc.

Signed: N.S. [Signature]

Date: 2nd May 2019

Part C

1.1 Identify the problem: (break down into smaller problems, write individual stages)

- Ask user to enter position of first card between 0-11 and store in card1
- Ask user to enter position of second card and store in card2
- Print on screen the position of first card and the number of first pair ad store in card1 and pairs[card1]
- Print on screen the position of the second card and the number of second pair and store in card2 and pairs[card2]
- If the pair matches print on screen it's a match
- Print on screen pairs found and the number of pairs found and store in found
- Loop back to the beginning and count turn
- If pairs don't match then print sorry no match
- Loop back to beginning and count turn
- If either card has already been picked print on screen sorry the position of this card has been removed
- Once all pairs have been found the loop will stop
- Print on screen you've found all the pairs
- Print on screen the total number of turns taken is and store in turns

1.2 Annotated code

```
#include <stdio.h>

void main()
{
    int pairs[12] = { 6,4,2,3,1,2,5,1,3,4,5,6 };
    int card1, card2;
    int found = 0;
    int turn = 0;

    while (found < 6)
    {
        turn += 1;

        printf("Enter position of first card", pairs[0 - 11]);
        scanf("%d", &card1);
        printf("Enter position of second card", pairs[0 - 11]);
        scanf("%d", &card2);
```

```

    if (pairs[card1] == 0)
    {
        printf("sorry the card for this position %d has been removed\n", card1);
    }
    else if (pairs[card2] == 0)
    {
        printf("sorry the card for this position %d has been removed\n", card2);
    }
    else if (pairs[card1] == pairs[card2])
    {
        found += 1;
        printf("first card choice %d = %d\n", card1, pairs[card1]);
        printf("second card choice %d = %d\n", card2, pairs[card2]);
        printf("its a match ");
        pairs[card1] = 0;
        pairs[card2] = 0;
        printf("pairs found =%d\n", found);
    }
    else
    {
        printf("first card choice %d = %d\n", card1, pairs[card1]);
        printf("second card choice %d = %d\n", card2, pairs[card2]);
        printf("sorry no match\n");
    }
}
printf("youve found all the pairs");
printf("the total number of turns is %d", turn);
}

```

1.3 Testing

1.3.1 Normal values

```
Enter position of first card0
Enter position of second card11
first card choice 0 = 6
second card choice 11 = 6
its a match pairs found =1
Enter position of first card1
Enter position of second card9
first card choice 1 = 4
second card choice 9 = 4
its a match pairs found =2
Enter position of first card2
Enter position of second card5
first card choice 2 = 2
second card choice 5 = 2
its a match pairs found =3
Enter position of first card7
Enter position of second card10
first card choice 7 = 1
second card choice 10 = 5
sorry no match
```

```
Enter position of first card3
Enter position of second card8
first card choice 3 = 3
second card choice 8 = 3
its a match pairs found =4
Enter position of first card4
Enter position of second card7
first card choice 4 = 1
second card choice 7 = 1
its a match pairs found =5
Enter position of first card7
Enter position of second card9
sorry the card for this position 7 has been removed
Enter position of first card6
Enter position of second card10
first card choice 6 = 5
second card choice 10 = 5
its a match pairs found =6
youve found all the pairs
the total number of turns is 8
```

Result: programme ran as expected and was successful. When the pairs were the same it printed "it's a match". When the pairs were different it printed "sorry no match". If a pair had already been picked it printed "sorry the card for this position has already been moved". Once all pairs were found it printed "you've found all the pairs". It calculated the amount of turns correctly and printed the total number of turns. Unfortunately the positions of the cards aren't randomised so they will be the same every game.

1.3.2 normal and negative values

```
Enter position of first card7
Enter position of second card-6
first card choice 7 = 1
second card choice -6 = -6
sorry no match
Enter position of first card-4
Enter position of second card-1
first card choice -4 = -858993460
second card choice -1 = -858993460
its a match pairs found =1
Enter position of first card
```

Result: as shown above a problem was encountered when using negative value. When a negative value was used as just a second choice the negative value was used as the pair resulting in no match as it was different to the first choice. When 2 negative values were used for the position choices the pairs came back the same but the values were completely different to pairs used in the game this however resulted in a match.

1.3.3 values higher than 11 and the same

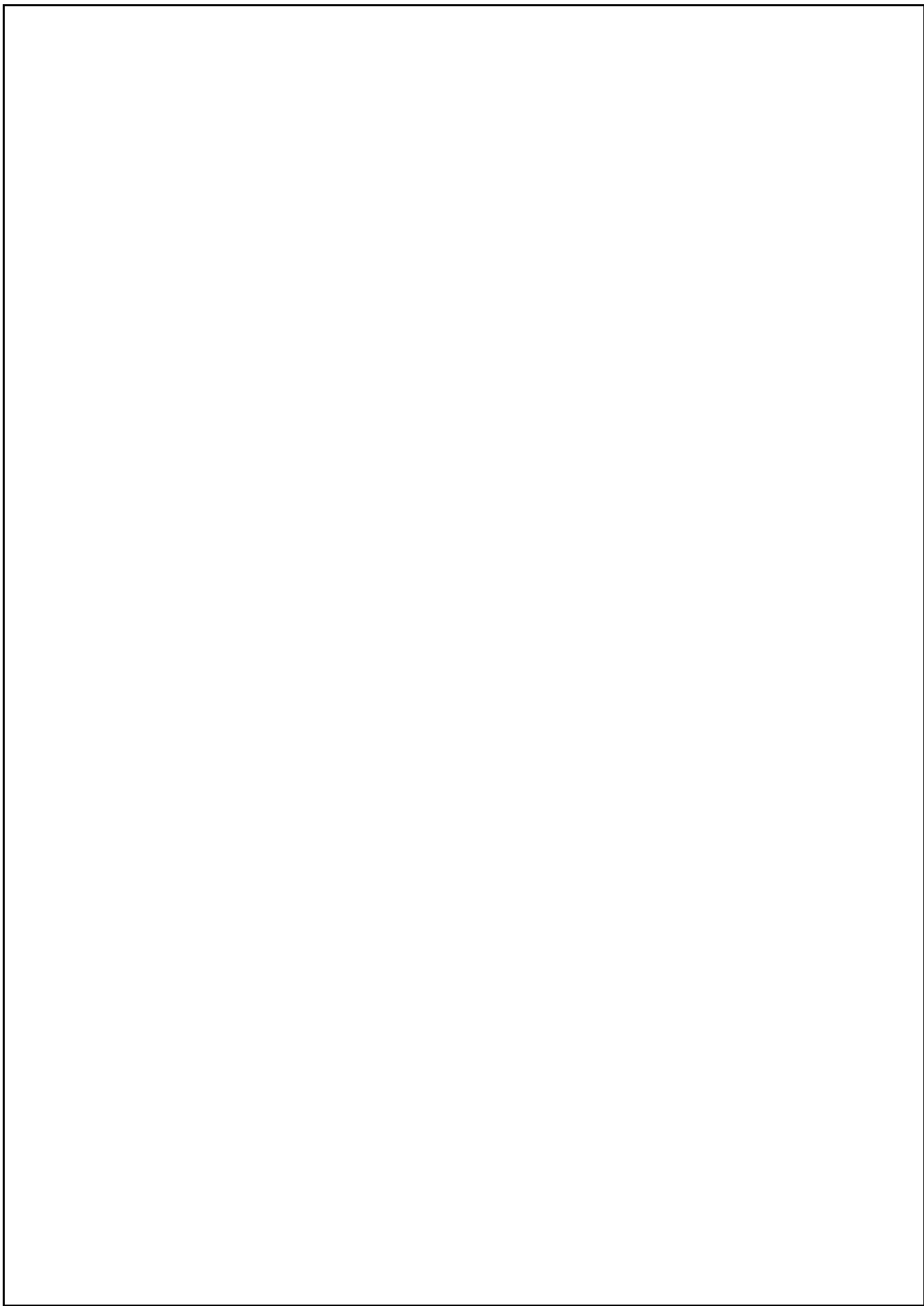
```
Enter position of first card5
Enter position of second card5
first card choice 5 = 2
second card choice 5 = 2
its a match pairs found =1
Enter position of first card12
Enter position of second card12
first card choice 12 = -858993460
second card choice 12 = -858993460
its a match pairs found =2
Enter position of first card14
Enter position of second card22
first card choice 14 = 5241980
second card choice 22 = 12653374
sorry no match
Enter position of first card
```

Result: As shown above there was a problem when using the same value for each card choice and a value higher than the value of positions in the game. When using the same value, the pairs were the same as the card has not yet been removed resulting in a match. When using the same value, but higher than the value of positions the pairs were the same, but the value of the pair was different to the pairs used in the game still resulting in a match. Also, when using to different values higher than the value of the positions the pairs were different and the values were different to the pairs used in the game resulting in no match.

1.3.4 letters

```
Enter position of first card  
Enter position of second card  
C:\Users\nickr\source\repos\Project18\Debug\Pro  
Press any key to close this window . . .
```

Result: as shown above when using letters the program would not run as it wouldn't let the user enter anything for the second choice.



C Programs and Documentation

GRADEMARK REPORT

FINAL GRADE

/100

GENERAL COMMENTS

Instructor

PART C

Coding/Annotation- A basic game, but no function work. You have not shuffled the cards, so it's the same game every time. Also, you need to think about validation - the use must be constrained to enter 0-11 for their card choice. They should not enter -1 or 12 for example. However, a working game overall 10/20

Documentation - Breakdown OK but you need to number the stages. Also, you should not reference code in your breakdown - just the logic. So you don't need to mention `paris[card1]` for example - use of an array is an implementation decision, this section is about the logic of the program. Testing is fairly extensive and you have rightly pointed out where there are deficiencies in the code (eg negative number) 9/15

Total 19/35

Overall total for assignment: 28.5+19

47.5/100

Test mark 71%

Overall mark for module 55%

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8
