

Aidan Griffiths Test sheet 4

by Aidan Griffiths

Submission date: 05-Jan-2026 03:23PM (UTC+0000)

Submission ID: 272155310

File name: 66131_Aidan_Griffiths_Aidan_Griffiths_Test_sheet_4_703116_1054074420.docx (222.48K)

Word count: 1391

Character count: 6987

Name: Aidan Griffiths

Student Number: 2501271

WORKSHEET 4 – TESTING (Dice Duel)

1. Original code:

```
#define _CRT_SECURE_NO_WARNINGS
#include <math.h>
#include <Stdio.h>
#include <stdlib.h>
#include <time.h>

void main()
{
    int myrand1, myrand2, myrand3, myrand4, myrand5, myrand6;
    int low = 1, high = 6;
    char choice;
    int usertotal = 0, comptotal = 0;
    int userscore = 0, compscore = 0;
    int stats[30];
    int round = 0;
    int i;

    printf("Welcome to the dice game!\n\n");
    printf("Are you ready to play? Y/N?\n");
    scanf(" %c", &choice);

    srand(time(NULL));

    while (choice == 'Y' || choice == 'y')
    {
        myrand1 = rand() % (high - low + 1) + low;
        myrand2 = rand() % (high - low + 1) + low;
        myrand3 = rand() % (high - low + 1) + low;
        myrand4 = rand() % (high - low + 1) + low;
        myrand5 = rand() % (high - low + 1) + low;
        myrand6 = rand() % (high - low + 1) + low;

        printf("You rolled...\n");
        printf("%d, %d, %d\n", myrand1, myrand2, myrand3);
        printf("Computer rolled...\n");
        printf("%d, %d, %d\n", myrand4, myrand5, myrand6);

        usertotal = myrand1 + myrand2 + myrand3;
        comptotal = myrand4 + myrand5 + myrand6;
```

```

        if (usertotal > comptotal)
        {
            userscore = userscore + 1;
            printf("Current scores are: You: %d Computer: %d\n\n",
userscore, compscore);
            stats[round] = 1;
        }
        else if (usertotal < comptotal)
        {
            compscore = compscore + 1;
            printf("Current scores are: You: %d Computer: %d\n\n",
userscore, compscore);
            stats[round] = 2;
        }
        else
        {
            printf("Current scores are: You: %d Computer: %d\n\n",
userscore, compscore);
            stats[round] = 3;
        }

        if (stats[round] == 1)
        {
            printf("You won!\n\n");
        }

        if (stats[round] == 2)
        {
            printf("Computer won!\n\n");
        }

        if (stats[round] == 3)
        {
            printf("DRAW!\n\n");
        }

        printf("Here are the statistics: \n");

        for (i = 0; i <= round; i++)
        {
            printf("Round %d: %d\n", i + 1, stats[i]);
        }

        if (userscore == 12)
        {
            printf("GAME OVER: YOU WIN!\n");
            scanf(" %c", &choice);
            printf("Play again?");
            if (choice == 'Y' || choice == 'y')
            {

```

```

        main();
    }
}
else if (compscore == 12)
{
    printf("GAME OVER: COMPUTER WINS!\n");
    printf("Play again?");
    scanf(" %c", &choice);
    if (choice == 'Y' || choice == 'y')
    {
        main();
    }
}

printf("Ready to play again? Y/N?\n\n");
scanf(" %c", &choice);

round = round + 1;
printf("***%d***", round);
}

printf("Okay. See you later...");
}

```

2. Code Testing

For each test below (i – ix), run your code and show the results of the scenarios below. Present a screenshot of what happened when the programme ran. Did the programme behave correctly or did it do something unexpected, and if so, why?

Part a)

- i. Computer wins round
- ii. User wins round
- iii. Draw
- iv. "Go again" – enter *y*
- v. "Go again" – enter *n*
- vi. "Go again" – enter *Y*
- vii. "Go again" – enter *b*

Part b)

- viii. Show score/score tracking (depending on how you have implemented your program)
- ix. Show an overall win (however this is achieved eg first to 5, best of 12 etc)

Part c)

x. Analyse your code and write a short paragraph commenting on the efficiency of the original code. Could efficiency be improved and if so comment on what is currently inefficient?

Template for tests:

Test: Computer wins round/User wins round
Screenshot(s) of output:

```
Welcome to the dice game!

Are you ready to play? Y/N?
y
You rolled...
6, 5, 4
Computer rolled...
1, 1, 2
Current scores are:  You: 1 Computer: 0

You won!

Here are the statistics:
Round 1: 1
Ready to play again? Y/N?

y
**1**You rolled...
2, 2, 1
Computer rolled...
6, 4, 2
Current scores are:  You: 1 Computer: 1

Computer won!

Here are the statistics:
Round 1: 1
Round 2: 2
Ready to play again? Y/N?
```

Result: Did the programme behave as expected ie syntactically and logically correct?

YES

If YES, comment on what happened.

If NO, what went wrong and why? Suggest how things could be improved

When the user or computer wins the round, the system adds 1 to the score of whoever won the round and produces the current score and the statistics of the game.

Test: Draw

Screenshot(s) of output:

```
**2**You rolled...
6, 4, 1
Computer rolled...
2, 5, 4
Current scores are:  You: 1 Computer: 1

DRAW!

Here are the statistics:
Round 1: 1
Round 2: 2
Round 3: 3
Ready to play again? Y/N?
```

Result: Did the programme behave as expected ie syntactically and logically correct?

YES

If YES, comment on what happened.

If NO, what went wrong and why? Suggest how things could be improved

The system correctly added the score and stated DRAW as the result. Because it was a draw, the score remains the same and nothing is added to the computer or user score.

Test: Go again Enter "y"

Screenshot(s) of output:

```

Here are the statistics:
Round 1: 2
Ready to play again? Y/N?

y
**1**You rolled...
4, 5, 5
Computer rolled...
2, 2, 4
Current scores are:  You: 1 Computer: 1

You won!

Here are the statistics:
Round 1: 2
Round 2: 1
Ready to play again? Y/N?

```

Result: Did the programme behave as expected ie syntactically and logically correct?

YES

If YES, comment on what happened.

If NO, what went wrong and why? Suggest how things could be improved

The system accepted "y" as an answer and continued with the game.

Test: Go again Enter "n"

Screenshot(s) of output:

```

Here are the statistics:
Round 1: 1
Ready to play again? Y/N?

n
**1**Okay. See you later...

```

Result: Did the programme behave as expected ie syntactically and logically correct?

YES

If YES, comment on what happened.

If NO, what went wrong and why? Suggest how things could be improved
The system accepted "n" as an answer and ended the game.

Test: Go again Enter "Y"

Screenshot(s) of output:

```
Ready to play again? Y/N?  
Y  
**1**You rolled...  
4, 4, 4  
Computer rolled...  
6, 5, 3  
Current scores are: You: 0 Computer: 2  
Computer won!
```

Result: Did the programme behave as expected ie syntactically and logically correct?
YES/NO (delete as appropriate)
If YES, comment on what happened.
If NO, what went wrong and why? Suggest how things could be improved
The system accepted "Y" as an answer and continued the game

Test: Go again

Screenshot(s) of output:

```
Ready to play again? Y/N?  
b  
**1**Okay. See you later...
```

Result: Did the programme behave as expected ie syntactically and logically correct?
YES
If YES, comment on what happened.
If NO, what went wrong and why? Suggest how things could be improved
The system accepts "b" as the user not wanting to continue and ends the game

Test: Score tracking

Screenshot(s) of output:

```
Current scores are:  You: 0 Computer: 3

Computer won!

Here are the statistics:
Round 1: 2
Round 2: 2
Round 3: 2
Ready to play again? Y/N?
```

Result: Did the programme behave as expected ie syntactically and logically correct?

YES

If YES, comment on what happened.

If NO, what went wrong and why? Suggest how things could be improved

The system record the current score after every round and outputs the score between the user and computer.

Test: Full game end

Screenshot(s) of output:

```
**15**You rolled...
2, 3, 2
Computer rolled...
6, 2, 5
Current scores are:  You: 3 Computer: 12

Computer won!

Here are the statistics:
Round 1: 2
Round 2: 2
Round 3: 2
Round 4: 2
Round 5: 2
Round 6: 2
Round 7: 2
Round 8: 1
Round 9: 1
Round 10: 2
Round 11: 3
Round 12: 2
Round 13: 1
Round 14: 2
Round 15: 2
Round 16: 2
GAME OVER: COMPUTER WINS!
Play again?
```

Result: Did the programme behave as expected ie syntactically and logically correct?

YES

If YES, comment on what happened.

If NO, what went wrong and why? Suggest how things could be improved

The game ended and the computer won the game with a score of 3:12. The system ends the game and offers the user to play again.

3. Consider the results of the above tests and your analysis of program efficiency and revisit your original code in order to address any tests that failed and/or improve code efficiency.

(Note: you may not be able to solve all of the problems that were encountered, but solve as many as you can.)

Paste your full revised code here :

```
#define _CRT_SECURE_NO_WARNINGS
#include <math.h>
#include <Stdio.h>
#include <stdlib.h>
#include <time.h>

void main()
{
    int myrand1, myrand2, myrand3, myrand4, myrand5, myrand6;
    int low = 1, high = 6;
    char choice;
    int usertotal = 0, comptotal = 0;
    int userscore = 0, compscore = 0;
    int stats[30];
    int round = 0;
    int i;

    printf("Welcome to the dice game!\n\n");
    printf("Are you ready to play? Y/N?\n");
    scanf(" %c", &choice);

    srand(time(NULL));

    while (choice == 'Y' || choice == 'y')
    {
        myrand1 = rand() % (high - low + 1) + low;
        myrand2 = rand() % (high - low + 1) + low;
        myrand3 = rand() % (high - low + 1) + low;
        myrand4 = rand() % (high - low + 1) + low;
        myrand5 = rand() % (high - low + 1) + low;
        myrand6 = rand() % (high - low + 1) + low;

        printf("You rolled...\n");
        printf("%d, %d, %d\n", myrand1, myrand2, myrand3);
        printf("Computer rolled...\n");
        printf("%d, %d, %d\n", myrand4, myrand5, myrand6);

        usertotal = myrand1 + myrand2 + myrand3;
        comptotal = myrand4 + myrand5 + myrand6;

        if (usertotal > comptotal)
        {
            userscore = userscore + 1;
        }
    }
}
```

```

        printf("Current scores are: You: %d Computer: %d\n\n", userscore,
compscore);
        stats[round] = 1;
    }
    else if (usertotal < comptotal)
    {
        compscore = compscore + 1;
        printf("Current scores are: You: %d Computer: %d\n\n", userscore,
compscore);
        stats[round] = 2;
    }
    else
    {
        printf("Current scores are: You: %d Computer: %d\n\n", userscore,
compscore);
        stats[round] = 3;
    }

    if (stats[round] == 1)
    {
        printf("You won!\n\n");
    }

    if (stats[round] == 2)
    {
        printf("Computer won!\n\n");
    }

    if (stats[round] == 3)
    {
        printf("DRAW!\n\n");
    }

    printf("Here are the statistics: \n");

    for (i = 0; i <= round; i++)
    {
        printf("Round %d: %d\n", i + 1, stats[i]);
    }

    if (userscore == 12)
    {
        printf("GAME OVER: YOU WIN!\n");
        scanf(" %c", &choice);
        printf("Play again?");
        if (choice == 'Y' || choice == 'y')
        {
            main();
        }
    }
    else if (compscore == 12)

```

```
        {
            printf("GAME OVER: COMPUTER WINS!\n");
            printf("Play again?");
            scanf(" %c", &choice);
            if (choice == 'Y' || choice == 'y')
            {
                main();
            }
        }

        printf("Ready to play again? Y/N?\n\n");
        scanf(" %c", &choice);

        round = round + 1;
        printf("***%d***", round);
    }

    printf("Okay. See you later...");
}
}
```

Aidan Griffiths Test sheet 4

GRADEMARK REPORT

FINAL GRADE

15/0

GENERAL COMMENTS

WORKSHEET 4

Coding- I'm a little confused with the statistics which say 1 or 2- translate these into "Round 1: Computer Won" etc "Round 1: 2" is a bit meaningless to the user.

Good to see use of array to hold the statistic, you could also have used an array for the random numbers to streamline things and avoid the need to copy and paste very similar lines of code- with an array you can also use loops to improve efficiency(eg print and add in the same loop) A good solution overall

7.5/10

Testing - I might validate for something that isn't y or n rather than ending the game on a b. Also the statistics are not very user-friendly as commented on above. Otherwise a good discussion.

8/10

Overall- 15.5/20

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
