

Test sheet 4

by Josh Willis

Submission date: 27-Dec-2025 11:55PM (UTC+0000)

Submission ID: 271789875

File name: 58111_Josh_Willis_Test_sheet_4_703116_592452467.docx (249.3K)

Word count: 1370

Character count: 6577

Name: Joshua Willis

Student Number: 2405267

WORKSHEET 4 – TESTING (Dice Duel)

1. Original code:

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <time.h>
#include <stdlib.h>

void main()
{
    int UserNum1, UserNum2, UserNum3, ComNum1, ComNum2, ComNum3;
    int high = 6, low = 1;
    int ComTot = 0, UserTot = 0;
    int UserScore = 0, ComScore = 0;
    char choice;
    int stats[23];
    int round = 0;

    srand(time(NULL));

    printf("are you ready to play? (Y/N)");
    scanf(" %c", &choice);

    while (choice == 'y' || choice == 'Y')
    {
        UserNum1 = rand() % (high - low + 1) + low;
        UserNum2 = rand() % (high - low + 1) + low;
        UserNum3 = rand() % (high - low + 1) + low;
        ComNum1 = rand() % (high - low + 1) + low;
        ComNum2 = rand() % (high - low + 1) + low;
        ComNum3 = rand() % (high - low + 1) + low;

        printf("you rolled...\n");
        printf("%d %d %d\n", UserNum1, UserNum2, UserNum3);

        printf("computer rolled...\n");
        printf("%d %d %d\n", ComNum1, ComNum2, ComNum3);

        ComTot = ComNum1 + ComNum2 + ComNum3;
        UserTot = UserNum1 + UserNum2 + UserNum3;

        printf("your roll was %d and the computer roll was %d\n", UserTot, ComTot);
```

```

    if (ComTot == UserTot)
    {
        printf("the game is a draw\n\n");
        stats[round] = 1;
    }
    else if (ComTot > UserTot)
    {
        printf("you lose\n\n");
        ComScore = ComScore + 1;
        stats[round] = 2;
    }
    else
    {
        printf("you win\n\n");
        UserScore = UserScore + 1;
        stats[round] = 3;
    }

    printf("current scores are: \nuser - %d  \nComputer - %d\n", UserScore, ComScore);
    printf("round stats:\n");
    round = round + 1;
    for (int i = 0; i < round; i++)
    {
        if(stats[i] == 1)
        {
            printf("round %d draw\n", i + 1);
        }
        else if (stats[i] == 2)
        {
            printf("round %d lose\n", i + 1);
        }
        else if (stats[i] == 3)
        {
            printf("round %d win\n", i + 1);
        }
    }

    if (ComScore == 12)
    {
        printf("GAME OVER: YOU LOSE");

    }
    else if (UserScore == 12)
    {
        printf("GAME OVER: YOU WIN");

    }

    printf("\n\nare you ready to play again? (Y/N)");

```

```
scanf("%c", &choice);  
}  
  
}
```

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?

Test:

Computer wins round

Screenshot(s) of output:

```
are you ready to play? (Y/N)y
you rolled...
2 4 1

computer rolled...
3 3 2

your roll was 7 and the computer roll was 8
you lose

current scores are:
user - 0
Computer - 1

round stats:
round 1 lose

are you ready to play again? (Y/N)|
```

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

YES

All programme worked as is expected and the layout of all the information is correct this shows correctly the computer won the round

Test:

User wins round

Screenshot(s) of output:

```
are you ready to play? (Y/N)y
you rolled...
6 1 6

computer rolled...
6 1 5

your roll was 13 and the computer roll was 12

you win

current scores are:
user - 1
Computer - 0

round stats:
round 1 win

are you ready to play again? (Y/N)|
```

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

YES

This programme worked logically and showed the correct result of the user winning the round and the prompted to continue to the next round of the game

Test:
draw

Screenshot(s) of output:

```
are you ready to play? (Y/N)y
you rolled...
6 3 4

computer rolled...
5 5 3

your roll was 13 and the computer roll was 13
the game is a draw

current scores are:
user - 0
Computer - 0

round stats:
round 1 draw

are you ready to play again? (Y/N)|
```

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

The programme came out as a draw for the players and did what is meant to happen by keeping the scores at 0 for each and not increasing unless a round is won all display is correct

Test:
"Go again" – enter y

Screenshot(s) of output:

```
are you ready to play again? (Y/N)y
you rolled...
2 5 1

computer rolled...
2 4 6

your roll was 8 and the computer roll was 12
you lose

current scores are:
user - 0
Computer - 1
```

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

Entering 'y' into the play again section worked exactly how expected with the programme continuing to the next round of the game

Test:

"Go again" – enter n

Screenshot(s) of output:

```
are you ready to play again? (Y/N)n  
  
C:\uni things\software deveolpment\assignment 1\worksheet 4\x64\Debug\worksh  
eet 4.exe (process 13908) exited with code 0 (0x0).  
To automatically close the console when debugging stops, enable Tools->Optio  
ns->Debugging->Automatically close the console when debugging stops.  
Press any key to close this window . . .|
```

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

YES

After inputting 'n' into the play again section the programme ended just as it is meant to

Test:

"Go again" – enter Y

Screenshot(s) of output:

```
are you ready to play again? (Y/N)Y  
  
C:\uni things\software deveolpment\assignment 1\worksheet 4\x64\Debug\worksheet 4.exe (process 2816) exited with code 0  
(0x0).  
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso  
le when debugging stops.  
Press any key to close this window . . .|
```

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

NO

The programme ended unexpectedly when entering this test. The code should have continued to the next round

Test:

"Go again" – enter B

Screenshot(s) of output:

```
are you ready to play again? (Y/N)B  
  
C:\uni things\software deveolpment\assignment 1\worksheet 4\x64\Debug\worksheet 4.exe (process 8500) exited with code 0  
(0x0).  
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso  
le when debugging stops.  
Press any key to close this window . . .|
```

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

YES

This result is what was expected as it isn't a y which continues the code to the next round, after entering the test the programme ended to be restarted

Test:
Score tracking

Screenshot(s) of output:

```
current scores are:  
user - 6  
Computer - 11  
  
round stats:  
round 1 win  
round 2 lose  
round 3 win  
round 4 win  
round 5 draw  
round 6 lose  
round 7 draw  
round 8 win  
round 9 lose  
round 10 lose  
round 11 lose  
round 12 lose  
round 13 lose  
round 14 win  
round 15 lose  
round 16 win  
round 17 lose  
round 18 lose  
round 19 draw  
round 20 lose
```

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

The score tracking part of the tests went exactly as it meant to in the display and the logic of it following all the rounds correctly and tracking the individual player score each time a round is won

Test:
Overall win

Screenshot(s) of output:

```
current scores are:
user - 12
Computer - 6

round stats:
round 1 win
round 2 win
round 3 win
round 4 win
round 5 win
round 6 win
round 7 draw
round 8 lose
round 9 win
round 10 win
round 11 win
round 12 lose
round 13 lose
round 14 lose
round 15 draw
round 16 draw
round 17 lose
round 18 draw
round 19 lose
round 20 draw
round 21 draw
round 22 win
round 23 win
round 24 win
GAME OVER: YOU WIN

are you ready to play again? (Y/N)
```

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

YES

Once a player reaches 12 rounds won the game ends as it is meant to. Notifying the player wins or loses against the computer and the prompts for another input to begin the games from the start or end the programme in its entirety

The efficiency of the code is that it is all looped so it can always continue not requiring the user to rerun the code to play each round. To improve the efficiency of this programme is to make the section split into individual functions also then tidying the code and offering an easier way to understand and read the code.

(Copy and paste box above for each of the rest of the tests)

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 :

```
[Copy and paste your full revised code here – note, do not use a screenshot]
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <time.h>
#include <stdlib.h>

void main()
{
    int UserNum1, UserNum2, UserNum3, ComNum1, ComNum2, ComNum3;
    int high = 6, low = 1;
    int ComTot = 0, UserTot = 0;
    int UserScore = 0, ComScore = 0;
    char choice;
    int stats[23];
    int round = 0;

    srand(time(NULL));

    printf("are you ready to play? (Y/N)");
    scanf(" %c", &choice);

    while (choice == 'y' || choice == 'Y')
    {
        UserNum1 = rand() % (high - low + 1) + low;
        UserNum2 = rand() % (high - low + 1) + low;
        UserNum3 = rand() % (high - low + 1) + low;
        ComNum1 = rand() % (high - low + 1) + low;
        ComNum2 = rand() % (high - low + 1) + low;
        ComNum3 = rand() % (high - low + 1) + low;

        printf("you rolled...\n");
        printf("%d %d %d\n\n", UserNum1, UserNum2, UserNum3);

        printf("computer rolled...\n");
        printf("%d %d %d\n\n", ComNum1, ComNum2, ComNum3);

        ComTot = ComNum1 + ComNum2 + ComNum3;
        UserTot = UserNum1 + UserNum2 + UserNum3;

        printf("your roll was %d and the computer roll was %d\n\n", UserTot, ComTot);

        if (ComTot == UserTot)
        {
```

```

printf("the game is a draw\n\n");
stats[round] = 1;
}
else if (ComTot > UserTot)
{
printf("you lose\n\n");
ComScore = ComScore + 1;
stats[round] = 2;
}
else
{
printf("you win\n\n");
UserScore = UserScore + 1;
stats[round] = 3;
}

printf("current scores are: \nuser - %d \nComputer - %d\n\n", UserScore, ComScore);
printf("round stats:\n");
round = round + 1;
for (int i = 0; i < round; i++)
{
if(stats[i] == 1)
{
printf("round %d draw\n", i + 1);
}
else if (stats[i] == 2)
{
printf("round %d lose\n", i + 1);
}
else if (stats[i] == 3)
{
printf("round %d win\n", i + 1);
}
}

if (ComScore == 12)
{
printf("GAME OVER: YOU LOSE");
}
else if (UserScore == 12)
{
printf("GAME OVER: YOU WIN");
}

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

```


Test sheet 4

GRADEMARK REPORT

FINAL GRADE

15/0

GENERAL COMMENTS

WORKSHEET 4

Coding- Generally good. You could have used an array for you random numbers which would allow you to loop and therefore use print and addition in the same loop (for example), this would also avoid copying and pasting similar lines of code. Good use of stats, but be clear-when playing it I was a bit confused, perhaps put "You won" or "Computer won" to be clearer - just a usability thing.

7.5/10

Testing - Don't forget to say why something happened. Why did the code end unexpectedly? I might also validate for B (ie not y or n)

7.5/10

Overall- 15/20

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

