

Worksheet 1

by Jacob Evans

Submission date: 09-Jan-2024 03:06PM (UTC+0000)

Submission ID: 221203637

File name: 50439_Jacob_Evans_Worksheet_1_541134_855824184.docx (147.31K)

Word count: 593

Character count: 2842

WORKSHEET 1 – TESTING (Using Variables)

1. Original code:

[Copy and paste your original code from Moodle into this box. Do not change it.]

```
int myAge;
char mySurname[15], nationalInsurance[10];

printf("Please enter your surname: ");
scanf("%s", &mySurname);
printf("Please enter your 9 character national insurance number: ");
scanf("%s", &nationalInsurance);
while ((strlen(nationalInsurance) > 10 || strlen(nationalInsurance) < 9))
{
    printf("Please enter a 9 character National Insurance: ");
    scanf("%s", &nationalInsurance);
}
printf("Please enter your age: ");
scanf("%d", &myAge);
while (myAge >= 110)
{
    printf("Yeah you're a bit old fella, try again: ");
    scanf("%d", &myAge);
}
while (myAge <= 1)
{
    printf("You're a young'un, please enter a valid age: ");
    scanf("%d", &myAge);
}
printf("The details you have inputted are:\nSurname: %s\nNational
Insurance: %s\nAge: %d", mySurname, nationalInsurance, myAge);
```

2. Code Testing

For each test below, run your code and enter the values provided. Provide a screenshot of the programme running. Did the programme behave correctly or did it do something unexpected? Fill in a box for each test below.

Surname:

- i. Enter *JONES*
- ii. Leave blank
- iii. Enter 123

National Insurance Number:

- i. Enter PE 25 34 12 B
- ii. Leave blank
- iii. Enter AB 11 22 33 44 55 66 77 (ie too long for National Insurance number format)

Age

- i. Enter 34
- ii. Enter 150
- iii. Enter -4
- iv. Enter a letter

Test:

Surname i. Enter JONES

Screenshot(s) of output:

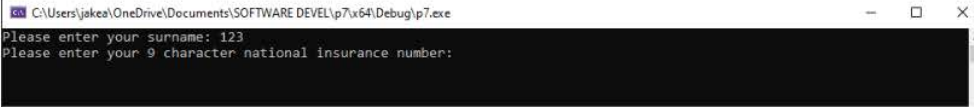


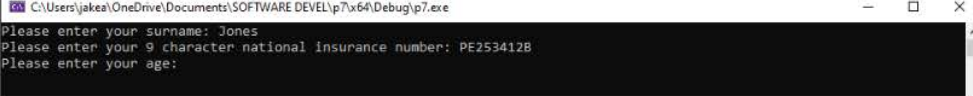
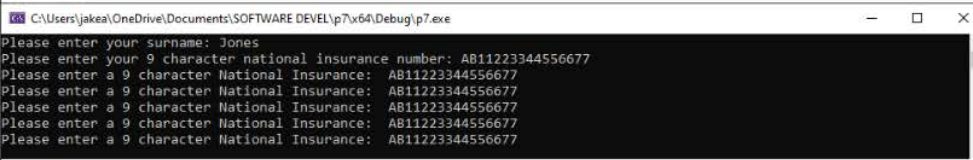
Result: Did the programme behave as expected ie syntactically and logically correct?

YES

Test: Surname ii Leave blank
Screenshot(s) of output: 
Result: Did the programme behave as expected ie syntactically and logically correct? NO (delete as appropriate) If NO, what went wrong?  1 The code does not proceed until the user inputs a value. This is not due to my code, but more to do with the control. I would need to change it so the user is notified to input data.

Continue the rest of the tests below. (Copy and paste a box for each one)

Test: Surname iii 123
Screenshot(s) of output: 
Result: Did the programme behave as expected ie syntactically and logically correct? NO (delete as appropriate) If NO, what went wrong? Code continues as if 123 is a correct input. I would need to change it so the user can only input a string and not an integer.

Test: National insurance i) PE 25 34 12 B
Screenshot(s) of output: 
Result: Did the programme behave as expected ie syntactically and logically correct? YES
Test: National insurance ii) blank
Screenshot(s) of output: 
Result: Did the programme behave as expected ie syntactically and logically correct? NO The code will not continue to the next step without the national insurance number. Entering space or just enter just drops a line in the control. This is "technically" correct as you shouldn't be allowed to skip this input, but it looks incorrect as I didn't take into account the "lack" of input.
Test: National insurance iii) AB 11 22 33 44 55 66 77
Screenshot(s) of output: 
Result: Did the programme behave as expected ie syntactically and logically correct? YES



3. Consider the results of the above tests and revisit your code in order to solve the problems.
(Note: you may not be able to solve all of them, but solve as many as you can.)

Paste your revised code here:

[Copy and paste your revised code here – note, do not use a screenshot]

```
int myAge;
char mySurname[15], nationalInsurance[10];

printf("Please enter your surname: ");
scanf("%s", &mySurname);
printf("Please enter your 9 character national insurance number: ");
scanf("%s", &nationalInsurance);
while ((strlen(nationalInsurance) > 10 || strlen(nationalInsurance) < 9))
{
    printf("Please enter a 9 character National Insurance: ");
    scanf("%s", &nationalInsurance);
}
printf("Please enter your age: ");
scanf("%d", &myAge);
{
    printf("That's not an integer, try again: ");
    scanf("%d", &myAge);
}
while (myAge >= 110)
{
    printf("Yeah you're a real old fella, try again: ");
    scanf("%d", &myAge);
}
while (myAge <= 1)
{
    printf("You're a young'un, please enter a valid age: ");
    scanf("%d", &myAge);
}

printf("The details you have inputted are:\nSurname: %s\nNational Insurance: %s\nAge: %d", mySurname,
nationalInsurance, myAge);
```

My revised code hasn't changed unfortunately. I attempted to resolve the testing for iv) which was entering a letter rather than an integer but could not get any headway. Fgets() was suggested with the Boolean numberRead=(scanf("%d", &myAge))==1) but it didn't seem to work. The same expression would be expected when testing for no input, as the expression above should advise the user to put an input in rather than leaving it blank.

Worksheet 1

GRADEMARK REPORT

FINAL GRADE

/10

GENERAL COMMENTS

Worksheet 1:

Coding: Careful with the age validation, the wrong age could still get through. (See my inline comment for why and how to fix it.)

You haven't made too much progress here, though you are close with the scanf amendment you suggest. Boolean is not a type in C, however.

You did address quite a bit of the validation in the original code, which is very good as you identified the possible input problems early.

3/5

Testing:

Good, you have understood this section well and suitably presented your results. 5/5

Total: 8/10

PAGE 1

PAGE 2

PAGE 3



Comment 1

Not necessarily wrong, it will continue until you enter something suitable, so no wrong values will get through

PAGE 4



Comment 2

Is this logically correct? the number is too long for a UK National Insurance number

PAGE 5



Comment 3

Amusing outputs :)

PAGE 6



Comment 4

Careful here. You could enter -4 and then 150 and still get through. Combine your while to `while(myAge>=110 || myAge<1)` to prevent this.

Worksheet 2 Testing

by Jacob Evans

Submission date: 03-Jan-2024 05:16PM (UTC+0000)

Submission ID: 220479937

File name: 50439_Jacob_Evans_Worksheet_2_Testing_541134_318431413.docx (68.76K)

Word count: 365

Character count: 1737

WORKSHEET 2 – TESTING (Carrots)

1. Original code:

[Copy and paste your original code from Moodle into this box. Do not change it.]

```
int width, length, total, carrotA = 64;
```

```
    printf("Please enter the length of the bed: ");
    scanf("%d", &length);
    while (length <= 0)
    {
        printf("You cannot have a negative length, please enter a valid length: ");
        scanf("%d", &length);
    }
    printf("Please enter the width of the bed: ");
    scanf("%d", &width);

    while (width <= 0)
    {
        printf("You cannot have a negative width, please enter a valid width: ");
        scanf("%d", &width);
    }

    total = length * width;
```

```
printf("Your total area bed is %d cm squared\n", total);
float totalTwo;
totalTwo = total % carrotA;
```

2. Code Testing

For each test below, run your code and enter the values provided. Provide a screenshot of the programme output. Did the programme behave correctly or did it do something unexpected? Fill in a box for each test below.

Input:

- i. Enter dimensions: Width 45 Length 85
- ii. Enter dimensions: Width 7 Length 25
- iii. Enter dimensions: Width -16 Length -20

Logic:

- i. Enter dimensions: Width 27 Length 30. The result should be 9 carrots.
- ii. Enter dimensions: Width 32 Length 32. The result should be 16 carrots.
- iii. Enter dimensions: Width 7 Length 16. The result should either be 0 carrots, or you may have accounted for this in your validation at the input phase.

Test:

Input - Enter dimensions: width 45 length 85

Screenshot(s) of output:



```
Microsoft Visual Studio Debug Console
Please enter the length of the bed: 45
Please enter the width of the bed: 85
Your total area bed is 3825 cm squared
C:\Users\Jakea\OneDrive\Documents\SOFTWARE DEVELOPMENT\Debug\PS.exe (process 30600) exited with code 0.
Press any key to close this window . . .
```

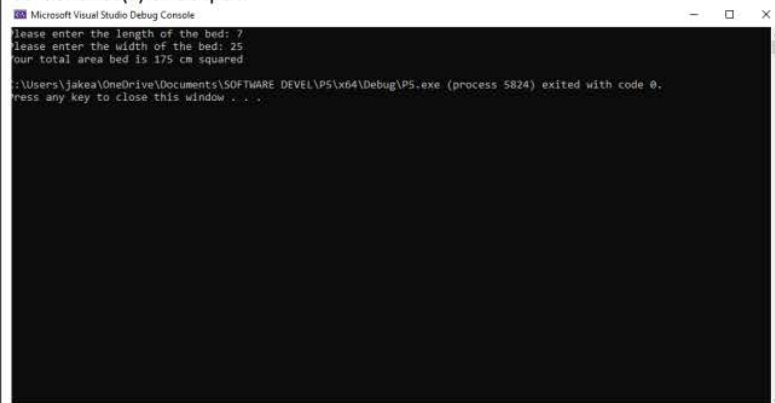
Result: Did the programme behave as expected ie syntactically and logically correct?

YES

Test:

Enter dimensions: width 7 length 25

Screenshot(s) of output:



```
Microsoft Visual Studio Debug Console
Please enter the length of the bed: 7
Please enter the width of the bed: 25
Your total area bed is 175 cm squared
C:\Users\jakea\OneDrive\Documents\SOFTWARE DEVELOPMENT\Debug\PS.exe (process 5824) exited with code 0.
Press any key to close this window . . .
```

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

No

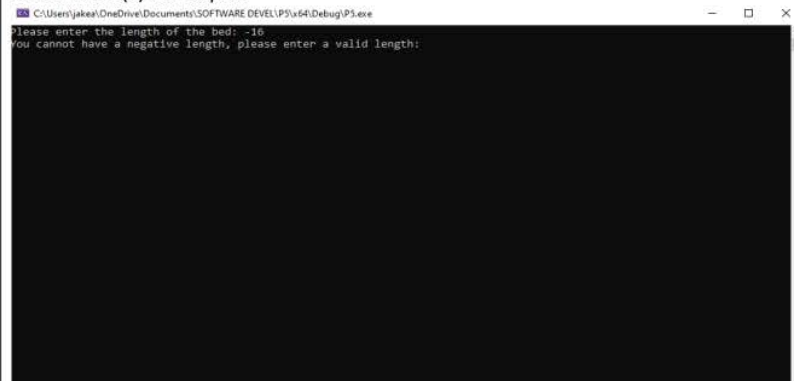
Code allows for carrot to be 7cm long, which is not ideal carrot growth length.

Continue the rest of the tests below. (Copy and paste a box for each one)

Test:

Negative length

Screenshot(s) of output:



```
C:\Users\jakea\OneDrive\Documents\SOFTWARE DEVELOPMENT\Debug\PS.exe
Please enter the length of the bed: -16
You cannot have a negative length, please enter a valid length:
```

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

YES.

I cannot answer the logic with the original code as I did not initially include the amount of carrots calculated, and since the code is incomplete.

3. Consider the results of the above tests and revisit your code in order to solve any problems discovered at the testing phase. (Note: you may not be able to solve all of them, but solve as many as you can.)

Paste your revised code here:

[Copy and paste your revised code here – note, do not use a screenshot]

```
int width, length, carrotA = 64;
float total;

printf("Please enter the length of the bed: ");
scanf("%d", &length);
while (length < 8)
{
    printf("You cannot grow a carrot with this length, try again: ");
    scanf("%d", &length);
}
printf("Please enter the width of the bed: ");
scanf("%d", &width);

while (width < 8)
{
    printf("You cannot grow a carrot with this width, try again: ");
    scanf("%d", &width);
}

total = (length * width) / carrotA;

int result = floor(total);

printf("You can grow %d amount of carrots", result);
```

This update code allows the user to input only carrot lengths and width greater than 8 (as carrots need at least an 8 x 8). I was initially so concentrated on the area of the carrot which wasn't useful.

The new code implemented also calculates the number of whole carrots (you cannot have 0.9 of a carrot) and displays it to the user.

The new code has resolved the earlier issues.

Worksheet 2 Testing

GRADEMARK REPORT

FINAL GRADE

/10

GENERAL COMMENTS

Worksheet 2:

Coding: Some progress has been made with the validation.

However, you have not addressed the logic of the number of carrots that can fit in the bed - each needs an 8 x 8 square. If you do length*width you get the total area of the bed. This will calculate that you can grow more carrots than you can in reality. You need to relate the carrot bed dimensions to the 8cm minimum dimensions on length and width. It's not enough to say carrot needs 64 - one of the 64s in total could be the sum of leftover strips of bed.

2.5/5

Testing:

Good as far as it goes - although appreciate you could not fully test initially.

4/5

Total: 6.5/10

Worksheet 3

by Jacob Evans

Submission date: 03-Jan-2024 05:27PM (UTC+0000)

Submission ID: 220480701

File name: 50439_Jacob_Evans_Worksheet_3_541134_1955492577.docx (127.88K)

Word count: 539

Character count: 2450

WORKSHEET 3 – TESTING (Arrays)

1. Original code:

[Copy and paste your original code from Moodle into this box. Do not change it.]

```
int i, j;
char product[5][15];
float price[5], total=0;

for (i = 0; i < 5; i++)
{
    printf("Enter product %d: ", i + 1);
    scanf("%s", &product[i]);
    printf("....and the price please: ");
    scanf("%f", &price[i]);
    if (price[i] < 0)
    {
        printf("Nothing in life is free, try again: ");
        scanf("%f", &price[i]);
    }
    total = (total + price[i]);
}

printf("\nProducts entered: \n");

for (j = 0; j < 5; j++)
{
    printf("%s \x9C%.2f\n", product[j], price[j]);
}

printf("\nTotal price is: \x9C%.2f", total);
total = (total * 0.85);
printf("\nDiscounted price is: \x9C%.2f", total);

}
```

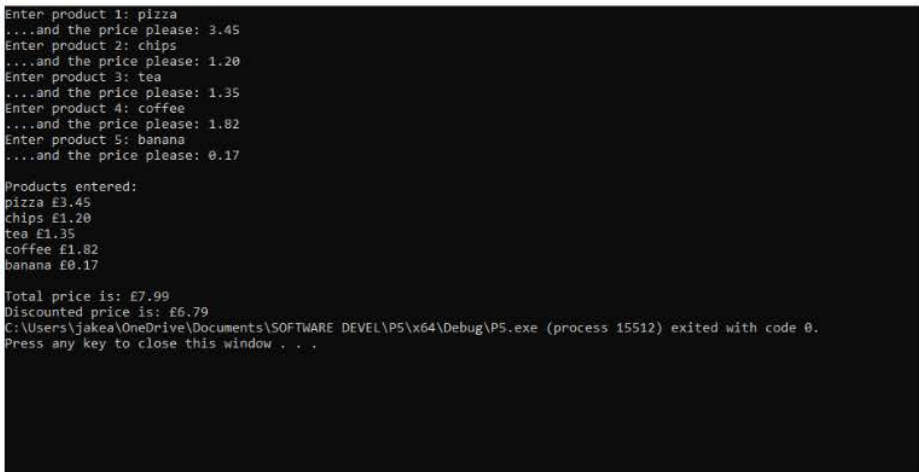
2. Code Testing

For each test below, run your code and enter the values provided. Provide a screenshot of the programme output. Did the programme behave correctly or did it do something unexpected? Fill in a box for each test below.

- i. Enter the 5 normal values as shown on the example screenshot (on the question sheet)
ie *pizza 3.45, chips 1.20, tea 1.35, coffee 1.82 and banana 0.17*
Double check the staff discount for the above values, is it correct?
- ii. Enter *pizza -2.34 chips -1.10, tea -1.22, coffee -2.05 and banana -0.18*
- iii. Enter your own values, but leave some of them blank

Test: i. Enter the 5 normal values as shown on the example
ie *pizza 3.45, chips 1.20, tea 1.35, coffee 1.82 and banana 0.17*
Double check the staff discount for the above values, is it correct?

Screenshot(s) of output:



```
Microsoft Visual Studio Debug Console
Enter product 1: pizza
....and the price please: 3.45
Enter product 2: chips
....and the price please: 1.20
Enter product 3: tea
....and the price please: 1.35
Enter product 4: coffee
....and the price please: 1.82
Enter product 5: banana
....and the price please: 0.17

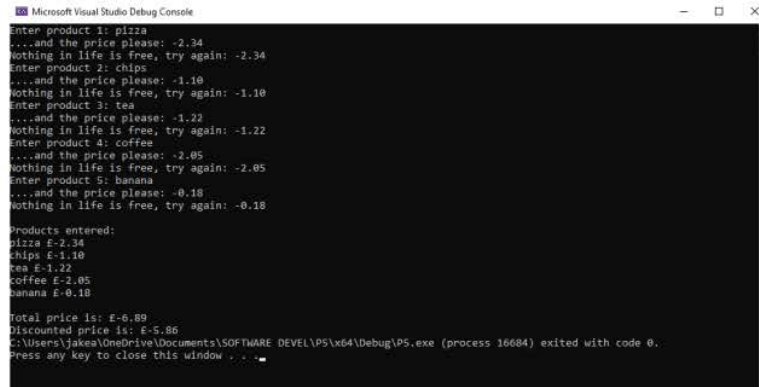
Products entered:
pizza £3.45
chips £1.20
tea £1.35
coffee £1.82
banana £0.17

Total price is: £7.99
Discounted price is: £6.79
C:\Users\jakea\OneDrive\Documents\SOFTWARE DEVEL\PS\x64\Debug\PS.exe (process 15512) exited with code 0.
Press any key to close this window . . .
```

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

Test: ii. Enter *pizza -2.34 chips -1.10, tea -1.22, coffee -2.05 and banana -0.18*

Screenshot(s) of output:



```
Microsoft Visual Studio Debug Console
Enter product 1: pizza
...and the price please: -2.34
Nothing in life is free, try again: -2.34
Enter product 2: chips
...and the price please: -1.10
Nothing in life is free, try again: -1.10
Enter product 3: tea
...and the price please: -1.22
Nothing in life is free, try again: -1.22
Enter product 4: coffee
...and the price please: -2.05
Nothing in life is free, try again: -2.05
Enter product 5: banana
...and the price please: -0.18
Nothing in life is free, try again: -0.18

Products entered:
pizza f-2.34
chips f-1.10
tea f-1.22
coffee f-2.05
banana f-0.18

Total price is: f-6.89
Discounted price is: f-5.86
C:\Users\jakea\OneDrive\Documents\SOFTWARE DEVELOPMENT\Debug\PS.exe (process 16684) exited with code 0.
Press any key to close this window . . .
```

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

NO

This is where the I had noticed the issues. Since my code only used an if loop, a user could bypass the code by inputting the price as negative again (as seen below) as it was only looped once.

Test: iii) Enter a few of your own values, leave some of the inputs blank

Screenshot(s) of output:



```
C:\Users\jakea\OneDrive\Documents\SOFTWARE DEVELOPMENT\Debug\PS.exe
Enter product 1: tea
...and the price please: 18
Enter product 2:
-
```

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

NO

My code does not allow a user to skip an item on the menu, suggesting that the person will always want 5 items (which shouldn't be the case, user should be allowed to input as many as they would like). The code also does not allow for any mistakes the user has made.

Any other tests you feel may be relevant, or reveal a strength or deficiency in your original code, please detail below:

Test:

Inputting a character instead of an integer for price

Screenshot(s) of output:

[illegible]

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

NO

Code breaks when inputting a character rather than an integer. If the user makes a mistake, then there this would be a major issue.

3. Consider the results of the above tests and revisit your code in order to solve any problems discovered at the testing phase. (Note: you may not be able to solve all of them, but solve as many as you can.)

Paste your revised code here:

[Copy and paste your revised code here – note, do not use a screenshot]

```
int j, items, i;
char product[5][15];
float price[5], total = 0;

printf("Welcome to the menu!\n");
printf("How many items have you got today?");
scanf("%d", &items);

for (i=0;i<items;i++)
{
    printf("Enter product %d: ", i + 1);
    scanf("%s", &product[i]);
    printf("....and the price please: ");
    scanf("%f", &price[i]);
    while (price[i] <= 0)
    {
        printf("Nothing in life is free, try again: ");
        scanf("%f", &price[i]);
    }
    total = (total + price[i]);
}

printf("\nProducts entered: \n");

for (j = 0; j < items; j++)
{
    printf("%s \x9C%.2f\n", product[j], price[j]);
}

printf("\nTotal price is: \x9C%.2f", total);
total = (total * 0.85);
printf("\nDiscounted price is: \x9C%.2f", total);
```

This is the new code I have come up with. I have solved the issue of looping only once when putting a negative value in (also 0, as nothing is free in life) by swapping out the "if" loop for a "while loop". I have also made it so the user can input the number of items they would like to buy, and the code loops for the amount. I was unsuccessful in allowing the user to skip inputs/make mistakes.

I am still unable to resolve the issue of missing an input, or inputting an incorrect variable (integer instead of a string and vice versa).

Worksheet 3

GRADEMARK REPORT

FINAL GRADE

/20

GENERAL COMMENTS

Worksheet 3:

Coding:

The use of while has improved things. I like the addition of the user stating the number of items, but be careful - if they put a number higher than 5 it will overflow the loop.

8/10

Testing:

You've got the main points here.

8/10

Total: 16/20

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

Worksheet 4

by Jacob Evans

Submission date: 03-Jan-2024 05:47PM (UTC+0000)

Submission ID: 220482048

File name: 50439_Jacob_Evans_Worksheet_4_541134_467099962.docx (127.28K)

Word count: 479

Character count: 2150

WORKSHEET 4 – TESTING (Arrays)

1. Original code:

[Copy and paste your original code from Moodle into this box. Do not change it.]

```
int high = 5, low = 1, points, random1, random2, random3, i;
char choice;
srand(time(NULL));
printf("Welcome to my program!\n\n");
printf("Let's go!\n\n");
printf("Would you like to spin (Y/N): ");
scanf(" %c", &choice);

while (choice == 'Y' || choice == 'y')
{
    random1 = rand() % (high - low + 1) + low;
    random2 = rand() % (high - low + 1) + low;
    random3 = rand() % (high - low + 1) + low;
    printf("%d %d %d\n", random1, random2, random3);

    if(random1<5 && random2<5 && random3<5)
    {
        if (random1 == random2 && random2 == random3)
        {
            printf("Nicely done you've earned 20 points!\nWant to play
again?(Y/N): ");
            scanf(" %c", &choice);
        }
    }
    else
    if(random1==5&&random1==random2 || random2==5&&random2==random3)
    {
        printf("Well done! You've earned 100 points!\n");
    }

    printf("Wanna play again ? (Y / N) : ");
    scanf(" %c", &choice);
}

printf("Why are you here?");
```

2. Code Testing

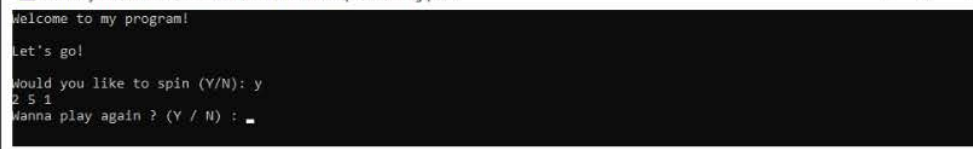
For each test below, run your code and enter the values provided. Provide a screenshot of the programme output. Did the programme behave correctly or did it do something unexpected? Fill in a box for each test below.

Test the following values for your fruit machine:

- i. 4 4 4 (20 points)
- ii. 1 3 2 (0 points)
- iii. 5 5 3 (100 points)
- iv. 5 2 5 (100 points)
- v. 1 5 5 (100 points)
- vi. 5 5 5 (2000 points)

vii. Enter "N" when user is asked whether they wish to make another spin

Test: i. 444 (20 points)
Screenshot(s) of output: 
Result: Did the programme behave as expected ie syntactically and logically correct? YES

Test: ii. 1 3 2 (0 points)
Screenshot(s) of output: 
Result: Did the programme behave as expected ie syntactically and logically correct? YES

Continue the rest of the tests below. (Copy and paste a box for each one)

Test:

iii. 5 5 3 winning

Screenshot(s) of output:

```
Wanna play again ? (Y / N) : y
5 5 3
Well done! You've earned 100 points!
Wanna play again ? (Y / N) :
```

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

YES

Test:

iv) 5 2 5 winning

Screenshot(s) of output:

```
Welcome to my program!
Let's go!
Would you like to spin (Y/N): y
2 5 1
Wanna play again ? (Y / N) : y
2 3 2
Wanna play again ? (Y / N) : y
3 1 1
Wanna play again ? (Y / N) : y
4 5 3
Wanna play again ? (Y / N) : y
5 1 2
Wanna play again ? (Y / N) : y
4 3 4
Wanna play again ? (Y / N) : y
5 2 5
Wanna play again ? (Y / N) :
```

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

NO

Code does not see 5 2 5 winning. I would need to change it so the code understands that 5 in more than one position is a win.

Test:

v) 1 5 5

Screenshot(s) of output:



```
C:\Users\jakee\OneDrive\Documents\SOFTWARE DEVELOPMENT\p7\p7.exe
Welcome to my program!

Let's go!

Would you like to spin (Y/N): Y
5 5 5
Well done! You've earned 100 points!
Wanna play again ? (Y / N) : Y
5 5 5
Well done! You've earned 100 points!
Wanna play again ? (Y / N) : Y
3 5 5
Well done! You've earned 100 points!
Wanna play again ? (Y / N) : _
```

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

Yes

Test:

vi) 5 5 5 Jackpot

Screenshot(s) of output:



```
Wanna play again ? (Y / N) : y
4 3 3
Wanna play again ? (Y / N) : y
5 4 1
Wanna play again ? (Y / N) : y
3 1 1
Wanna play again ? (Y / N) : y
5 5 5
Well done! You've earned 100 points!
Wanna play again ? (Y / N) : y
```

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

No

Game recognises as a win, but not as a jackpot.

Any other tests you feel may be relevant, or reveal a strength or deficiency in your original code, please detail below:

Test:

Different inputs

Screenshot(s) of output:

A screenshot of a terminal window with a black background and white text. The text shows the program's execution flow: a welcome message, a prompt to spin, a series of spins (5, 5, 5) each resulting in 100 points and a prompt to play again, and finally an exit message. The terminal path is C:\Users\jakea\OneDrive\Documents\SOFTWARE DEVEL\p7\x64\Debug\p7.exe.

```
Welcome to my program!
Let's go!
Would you like to spin (Y/N): Y
5 5 5
Well done! You've earned 100 points!
Wanna play again ? (Y / N) : Y
5 5 5
Well done! You've earned 100 points!
Wanna play again ? (Y / N) : Y
3 5 5
Well done! You've earned 100 points!
Wanna play again ? (Y / N) : h
Why are you here?
C:\Users\jakea\OneDrive\Documents\SOFTWARE DEVEL\p7\x64\Debug\p7.exe (process 16856) exited with code 0.
Press any key to close this window . . .
```

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

YES

Code only accepts Y or y as a valid option. Everything else ends the program.



3. Consider the results of the above tests and revisit your code in order to solve any problems discovered at the testing phase. (Note: you may not be able to solve all of them, but solve as many as you can.)

Paste your revised code here:

I have since revisited the code, and updated to not only look tidier but also included the jackpot winnings as well as fixed winning 100 points on 5 1 5 (random 1 and 3 being 5).

I could have added something that calculates the score as you take turns, as well as added a counter to see how many turns it took to get to a certain score within a certain amount of turns to make it a bit more complicated, but the code certainly answers the task.

[Copy and paste your revised code here – note, do not use a screenshot]

```
{
    int high = 5, low = 1, points, random1, random2, random3, i;
    char choice;
    srand(time(NULL));
    printf("Welcome to my program!\n\n");
    printf("Let's go!\n\n");
    printf("Would you like to spin (Y/N): ");
    scanf("%c", &choice);

    while (choice == 'Y' || choice == 'y')
    {
        random1 = rand() % (high - low + 1) + low;
        random2 = rand() % (high - low + 1) + low;
        random3 = rand() % (high - low + 1) + low;
        printf("%d %d %d\n", random1, random2, random3);

        if (random1 < 5 && random2 < 5 && random3 < 5)
        {
            if (random1 == random2 && random2 == random3)
            {
                printf("Nicely done you've earned 20 points!\n");
            }

            else if (random1 == random2 && random2 == random3)
            {
                printf("Jackpot! Well done, you've won 2000 points!\n");
            }
            else if (random1 == 5 && random1 == random2 || random2 == 5 && random2 == random3 ||
random1 == 5 && random1 == random3)
            {
                printf("Well done, you've won 100 points!\n");
            }
            printf("Wanna play again ? (Y / N) : ");
            scanf("%c", &choice);
        }

        printf("Why are you here?");
    }
}
```

Worksheet 4

GRADEMARK REPORT

FINAL GRADE

/20

GENERAL COMMENTS

Worksheet 4:

Coding:

A cumulative score is a harder prospect and would have scored more, nevertheless, a good attempt. Some of the code is not entirely efficient

6/10

Testing:

Tests are presented well with good discussion 10/10

Total: 16/20

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5



Comment 1

A bit harsh! :)

PAGE 6

Worksheet 5

by Jacob Evans

Submission date: 03-Jan-2024 06:15PM (UTC+0000)

Submission ID: 220483841

File name: 50439_Jacob_Evans_Worksheet_5_541134_621768568.docx (138.96K)

Word count: 663

Character count: 2979

WORKSHEET 5 – TESTING (Treasure Hunt)

For this worksheet, there are several possible approaches to the problem.

Create your own test set to show the strengths and/or limitations of your solution. Use as many of the boxes below as you need.

Test:

Input values larger than 15 or less than 0

Screenshot(s) of output:



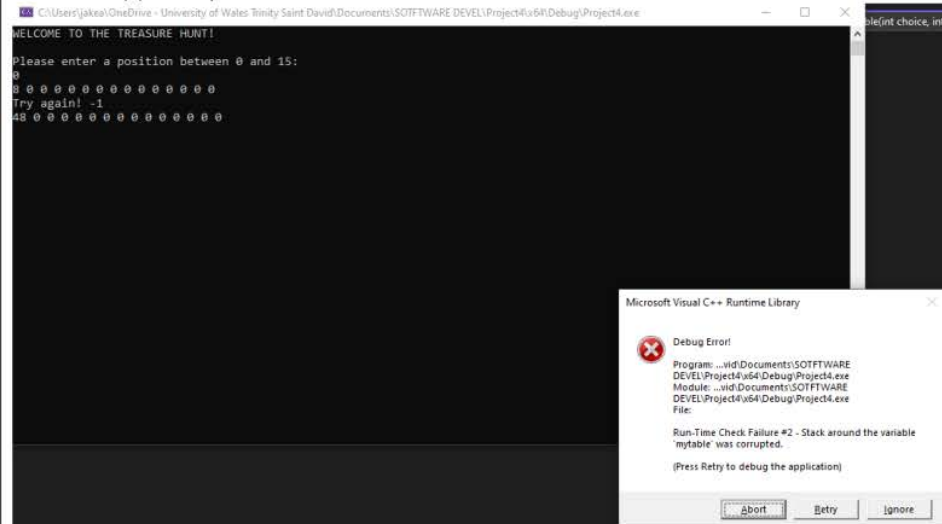
```
C:\Users\jakea\OneDrive - University of Wales Trinity Saint David\Documents\SOFTWARE DEVELOPMENT\Project4\Debug\Project4.exe
WELCOME TO THE TREASURE HUNT!
Please enter a position between 0 and 15:
-1
Invalid, please try again:
```

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

YES

Test:
Input values larger than 15 or less than 0 after returning table

Screenshot(s) of output:



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

NO

Error occurs after inputting a negative value once the table has been created. I believe that the code is attempting to re-run the while loop but hits a wall due to the created gameboard/table.

Test:
Creating the table

Screenshot(s) of output:



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

ISH and prompts the user to try again since they missed.

Issue – 48 is the first value of the array due to ACSII but I'm unsure how to solve this issue. I'm led to believe its an overflow.

Test:
Creating the table

Screenshot(s) of output:



Result: Did the programme behave as expected ie syntactically and logically correct?
YES and prompts the user to try again since they missed.

Test:
Winning

Screenshot(s) of output:



Result: Did the programme behave as expected ie syntactically and logically correct?
YES – Code turns green when the user inputs the correct position, and let's the user know how many tries.

Issue – I could not get the correct array to display "X". X is the conventional way of showing a hit, but not a major issue as I can display a hit as whatever number I'd like (666).

The gameboard also wouldn't return an updated version, therefore the miss is only applied on the chosen slot and never seen again.

Test:
Random input

Screenshot(s) of output:

Result: Did the programme behave as expected ie syntactically and logically correct?
NO
The code hates any input that isn't an integer. I'm still very unsure on how to limit the user to only integers.

This is the code I managed to make. The code allows a user to input a number between 0 and 15 to guess a position for the treasure hunt. The code does not allow for the user to input a number less than 0 and larger than 15 (control turns red and states invalid, and to try again). The code then prints the visual table (placed in table.h). The table shows an 8 as a miss (table does not update all attempts) and a 666 as a hit (winner). The screen then turns green when the user guesses correctly. I also added a counter which shows the user how many attempts they took.

I have tested this several times and the number  1 dom in all attempts.

I came across some issues when making and testing this code. The table initially prints the first array as 48 rather than a 0.  2 h makes sense in ASCII but I couldn't work out why it was doing this. I initially thought it was an overflow but doesn't look to be the case.

I also couldn't work out how to show an updated version of the gameboard/table. The code shows the miss as an 8 but that 8 does not remain on the gameboard. I believe it to be the loop in the code is resetting the board and only making the miss player choice an 8.

I also couldn't work out how to show a hit as an X. I opted to show a hit as a "666" as it still shows a win, but it isn't as "conventional" as an X.

I also noticed while testing that the console breaks when putting a -1 or larger than 15 number after the putting a correct position in. This is most likely due to the table being produced after skipping this loop.

Final code:

[Copy and paste your code here – note, do not use a screenshot]

Source(1)

```
void main()
{
    srand(time(NULL));
    int i, high = 15, low = 0, choice, myrand, counter=0;
    myrand = rand() % (high - low + 1) + low;
    printf("WELCOME TO THE TREASURE HUNT!\n");
    printf("\nPlease enter a position between 0 and 15: \n");
    scanf("%d", &choice);
    while (choice < 0 || choice>15)
    {
        system("color 40");
        printf("Invalid, please try again: ");
        scanf("%d", &choice);
        counter++;
    }

    while (choice != myrand)
    {
        table(choice, myrand);
        printf("\nTry again! ");
        scanf("%d", &choice);
        counter++;
    }
    system("color 2D");
    table(choice, myrand);
    printf("\nNicely done!!!\nIt took %d tries", counter);
}
```

Table.h

```
void table(int choice, int myrand)
{
    int mytable[15] = { '0' };
    if (choice != myrand)
    {
        for (int i = 0; i < 15; i++)
        {
            mytable[choice] = 8;
            printf("%d ", mytable[i]);
        }
    }
    else
    {
        for (int i = 0; i < 15; i++)
        {
            myrand = i;
            printf("%d ", mytable[i]);
            mytable[choice] = 666;
        }
    }
    Return mytable[choice];
}
```

Worksheet 5

GRADEMARK REPORT

FINAL GRADE

/30

GENERAL COMMENTS

Worksheet 5:

Coding: You cannot use X because your array is of integer type. You could use a character array, or using what you've done with 666 you could use an if statement - `if table[choice]==666 {printf("X");}`

A good attempt on the functions, but be careful - I don't think your return is being received in main.

Overall the code is a bit mixed up - though some of the elements are complete. 7/15

Testing: An honest assessment of the program produced. Since the programme is not complete, you could not fully test everything.

8/15

Total: 15/30

Organisation of report, writing style, presentation of work:

Generally good. Avoid the use of "I" in academic reports, instead use the *passive voice*. I've put a few examples inline to show you.

7/10

Overall total for assignment: 68.5%

Test mark: 77%

Overall mark for module: 71%

You've shown some good skills in the assignment - keep practising in the time between now and Year 2 to maintain/improve your skills.

PAGE 1

PAGE 2

PAGE 3

PAGE 4



Comment 1

Passive voice: "This has been tested several times and the number was random in all attempts"



Comment 2

Passive voice: Some issues were encountered when making and testing this code

PAGE 5



Comment 3

no comment



Comment 4

Not sure where this return is being received? Also you would need int table not void table to accommodate a return