

Kallem G Thomas - 2304300

by Kallem Thomas

Submission date: 02-Jan-2024 03:20PM (UTC+0000)

Submission ID: 220389166

File name: 49116_Kallem_Thomas_Kallem_G_Thomas_-_2304300_541134_1929971689.docx (134.74K)

Word count: 803

Character count: 4217

WORKSHEET 1 – TESTING (Using Variables)

1. Original code:

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

int Area[16] = { 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0 };

#include "Header.h"

void main()
{
    int age;
    char surname[20], nationalNum[10];

    printf("Please enter your surname: ");
    scanf("%s", &surname);
    while (strlen(surname) > 20 || strlen(surname) < 2)
    {
        if (strlen(surname) <= 1)
        {
            printf("Surely your surname isn't one letter, ");
        }
        else
        {
            printf("You have entered too many characters, ");
        }

        printf("Please try again: ");
        scanf("%s", &surname);
    }

    printf("Please enter your National insurance number: ");
    scanf("%s", &nationalNum);

    while (strlen(nationalNum) >= 10 || strlen(nationalNum) <= 8)
    {
        printf("You have not entered correct information, Please try again: ");
        scanf("%s", &nationalNum);
    }

    printf("Please enter your Age: ");
    scanf("%d", &age);

    while (age >= 130 || age <= 0)
    {
        printf("Surely you aren't %d\nPlease input your ACTUAL age: ", age);
        scanf("%d", &age);
    }

    printf("\n%s\n", surname);
}
```

```

    for (int i = 0; i < 9; i++)
    {
        if (i % 2 == 0)
        {
            printf(" ");
        }
        printf("%0c", nationalNum[i]);
    }

    printf("\n%d", age);
}

```

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:

```

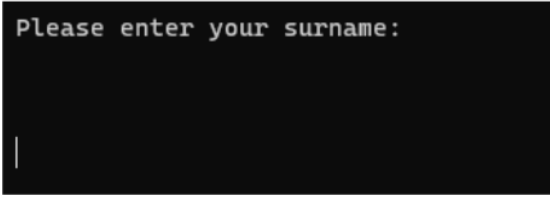
Please enter your surname: JONES
Please enter your National insurance number: pk343434c
Please enter your Age: 18

JONES
pk 34 34 34 c
18

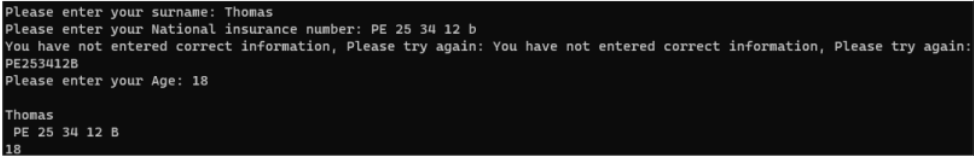
```

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. It does not allow the user to continue until they have given a surname. 

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

Test: iv. Enter PE 25 34 12 B
Screenshot(s) of output: 
Result: Did the programme behave as expected ie syntactically and logically correct? NO. The system built doesn't allow spaces to be entered.

Test:

v. Enter PE 25 34 12 B



Screenshot(s) of output:

```
Please enter your surname: Thomas
Please enter your National insurance number: PE 25 34 12 b
You have not entered correct information, Please try again: You have not entered correct information, Please try again:
PE253412B
Please enter your Age: 18

Thomas
PE 25 34 12 B
18
```

Result: Did the programme behave as expected ie syntactically and logically correct?
NO. The system built doesn't allow spaces to be entered.

Test:

vi. Leave blank

Screenshot(s) of output:

```
Please enter your surname: 123
Please enter your National insurance number: |
```

Result: Did the programme behave as expected ie syntactically and logically correct?
YES because you should not be able to continue without entering details.

Test:

iv. Enter AB 11 22 33 44 55 66 77 (ie too long for National Insurance number format)

Screenshot(s) of output:

```
Please enter your surname: thomas
Please enter your National insurance number: pk3423454332456778b
You have not entered correct information, Please try again: |
```

Result: Did the programme behave as expected ie syntactically and logically correct?
YES because you should not be able to continue without entering valid details.

Test:

v. Enter AB 11 22 33 44 55 66 77 (ie too long for National Insurance number format)

Screenshot(s) of output:

```
Please enter your surname: thomas
Please enter your National insurance number: pk3423454332456778b
You have not entered correct information, Please try again: |
```

Result: Did the programme behave as expected ie syntactically and logically correct?
YES because you should not be able to continue without entering valid details.

Age

v. Enter 34

```
Please enter your surname: thomas
Please enter your National insurance number: pk3423454332456778b
You have not entered correct information, Please try again: pk343434c
Please enter your Age: 34

thomas
pk 34 34 34 c
34
```

Yes, because the age is within valid range.

vi. Enter 150

```
Please enter your surname: Thomas
Please enter your National insurance number: pk232323c
Please enter your Age: 150
Surely you aren't 150
Please input your ACTUAL age: |
```

Yes because living to 150 isn't within range nor is it reality.

vii. Enter -4

```
Please input your ACTUAL age: -4
Surely you aren't -4
Please input your ACTUAL age: |
```

Yes because living to -4 isn't within range nor is it reality.

Enter a letter

```
Please input your ACTUAL age: Surely you aren't -858993460
Please input your ACTUAL age: Surely you aren't -858993460
Please input your ACTUAL age: Surely you aren't -858993460
Please input your ACTUAL age: Surely you aren't -858993460
Please input your ACTUAL age: Surely you aren't -858993460
Please input your ACTUAL age: Surely you aren't -858993460
Please input your ACTUAL age: Surely you aren't -858993460
Please input your ACTUAL age: Surely you aren't -858993460
Please input your ACTUAL age: Surely you aren't -858993460
Please input your ACTUAL age: Surely you aren't -858993460
Please input your ACTUAL age: Surely you aren't -858993460
Please input your ACTUAL age: Surely you aren't -858993460
Please input your ACTUAL age: Surely you aren't -858993460
Please input your ACTUAL age: Surely you aren't -858993460
Please input your ACTUAL age: Surely you aren't -858993460
Please input your ACTUAL age: Surely you aren't -858993460
Please input your ACTUAL age: Surely you aren't -858993460
```

No. The user should not be able to enter a letter but corrupts the system due to an integer not being able to store a character.

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:

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

```
void main()
{
```

```
    int age;
    char surname[20], nationalNum[10], userEnteredAge;
```

```
    printf("Please enter your surname: ");
    scanf("%s", &surname);
```

```
    while (strlen(surname) > 20 || strlen(surname) < 2)
    {
```

```
        if (strlen(surname) <= 1)
```

```
        {
```

```
            printf("Surely your surname isn't one letter, ");
```

```
        }
```

```
        else
```

```
        {
```

```
            printf("You have entered too many characters, ");
```

```
        }
```

```
        printf("Please try again: ");
        scanf("%s", &surname);
```

```
    }
```

```

printf("Please enter your National insurance number: ");
scanf("%s", &nationalNum);

while (strlen(nationalNum) >= 10 || strlen(nationalNum) <= 8)
{
    printf("You have not entered correct information, Please try again: ");
    scanf("%s", &nationalNum);
}

printf("Please enter your Age: ");
scanf("%s", &userEnteredAge);
age = atoi(&userEnteredAge);

while (age >= 130 || age <= 0)
{
    printf("Surely you aren't ' %s ' years old,\nPlease input your ACTUAL age: ",
&userEnteredAge);
    scanf("%s", &userEnteredAge);
    age = atoi(&userEnteredAge);
}

printf("\n%s\n", surname);

for (int i = 0; i < 9; i++)
{
    if (i % 2 == 0)
    {
        printf(" ");
    }
    printf("%c", nationalNum[i]);
}

printf("\n%d", age);

```

FINAL GRADE

/10

GENERAL COMMENTS

Worksheet 1:

Coding: You had a good amount of validation in your original code. An interesting approach to resolving difficulties with age testing but nevertheless works OK. If you were going to implement this in a live system, you would need to look at the format of the National Insurance number itself, rather than just the length.

4/5

Testing:

Good, you have understood this section well. I think there was a bit of copy and paste failure for some of it with some of the tests repeated.

4.5/5

Total: 8.5/10

PAGE 1

PAGE 2

PAGE 3

PAGE 4



Comment 1

Technically it's OK, because user can't get any further until they've engaged with the program

PAGE 5



Comment 2

Repeated?

PAGE 6

PAGE 7

PAGE 8

kallem g thomas - 2304300

by Kallem Thomas

Submission date: 05-Jan-2024 02:17PM (UTC+0000)

Submission ID: 220656346

File name: 49116_Kallem_Thomas_kallem_g_thomas_-_2304300_541134_1536053029.docx (57.17K)

Word count: 578

Character count: 3198

WORKSHEET 2 – TESTING (Carrots)

1. Original code:

```
#define CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <time.h>

void main()
{
    char userEnteredHeight, userEnteredlength;
    int total, totalRoom, length, height;

    printf("please enter the total length of your bed: ");
    scanf("%s", &userEnteredlength);
    height = atoi(&userEnteredlength);

    printf("please enter the total height of your bed: ");
    scanf("%s", &userEnteredHeight);
    length = atoi(&userEnteredHeight);

    totalRoom = (length / 8) * (height / 8);

    printf("You can grow a total of %d Carrots\n", totalRoom);

    if (totalRoom < 1)
    {
        printf("You cannot grow any carrots with your current bed.");
    }
}
```

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:

```
please enter the total length of your bed: 45
please enter the total height of your bed: 85
You can grow a total of 50 Carrots
```

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

YES. $45/8 = 5.625$ $85/8 = 10.625$ $5*10 = 50$.

Test: Enter dimensions: width 7 length 25
Screenshot(s) of output: <pre> please enter the total length of your bed: 7 please enter the total height of your bed: 25 You can grow a total of 0 Carrots You cannot grow any carrots with your current bed. </pre>
Result: Did the programme behave as expected ie syntactically and logically correct? YES. Doesnt matter how much bigger one is than the other, if either length or height is less than 8 it will not be able to grow a carrot as the requirements are 8x8.

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

Test: -16 -20
Screenshot(s) of output: <pre> please enter the total length of your bed: -16 please enter the total height of your bed: -20 You can grow a total of 4 Carrots </pre>
Result: Did the programme behave as expected ie syntactically and logically correct? NO. You cannot have a negative amount of space.

Test: 27 30

Screenshot(s) of output:

```
please enter the total length of your bed: 27  
please enter the total height of your bed: 30  
You can grow a total of 9 Carrots
```

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

Test: 32 32

Screenshot(s) of output:

```
please enter the total length of your bed: 32  
please enter the total height of your bed: 32  
You can grow a total of 16 Carrots
```

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

Test: 7 16

Screenshot(s) of output:

```
please enter the total length of your bed: 7  
please enter the total height of your bed: 16  
You can grow a total of 0 Carrots
```

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 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:

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

void main()
{

    char userEnteredHeight, userEnteredLength;
    int total, totalRoom, length, height;

    printf("please enter the total length of your bed: ");
    scanf("%s", &userEnteredLength);
    height = atoi(&userEnteredLength);

    while (height <= 0)
    {
        printf("That's not valid, please try again.\n");
        printf("please enter the total length of your bed: ");
        scanf("%s", &userEnteredLength);
        height = atoi(&userEnteredLength);
    }

    printf("please enter the total height of your bed: ");
    scanf("%s", &userEnteredHeight);
    length = atoi(&userEnteredHeight);

    while (length <= 0)
    {
        printf("That's not valid, please try again.\n");
        printf("please enter the total height of your bed: ");
        scanf("%s", &userEnteredHeight);
        height = atoi(&userEnteredHeight);
    }

    totalRoom = (length / 8) * (height / 8);

    printf("You can grow a total of %d Carrots\n", totalRoom);

    if (totalRoom < 1)
    {
        printf("You cannot grow any carrots with your current bed.");
    }
}
```

FINAL GRADE

/10

GENERAL COMMENTS

Worksheet 2:

Coding:

The width and length validation could actually be less than 8 - which is the minimum, not 0.

You have made good progress on the validation, could also consider an upper limit.

4/5

Testing:

Good presentation of results. 5/5

Total: 9/10

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5



Comment 1



Comment 2

8

Kallem G Thomas - 2304300

by Kallem Thomas

Submission date: 08-Jan-2024 03:07PM (UTC+0000)

Submission ID: 221013501

File name: 49116_Kallem_Thomas_Kallem_G_Thomas_-_2304300_541134_650045606.docx (52.35K)

Word count: 556

Character count: 2772

WORKSHEET 3 – TESTING (Arrays)

1. Original code:

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

char product[5][15];
float price[5], total = 0, tsd;

for (int i = 0; i < 5; i++)
{
    printf("Enter product %d: ", i + 1);
    scanf("%s", &product[i]);
    printf(".... and the price please: ");
    scanf("%f", &price[i]);

    while (price[i] < 0) //this is so that we dont give money away. there can still be free products
    {
        printf("Re-enter the price please: ");
        scanf("%f", &price[i]);
    }
}

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

for (int j = 0; j < 5; j++)
{
    printf("%s ", product[j]);
    printf("\x9c%.2f\n", price[j]);
    total = total + price[j];
}

printf("Total price is \x9c%.2f", total);

tsd = total * 0.15;
tsd = total - tsd;

printf("Total with staff discount is \x9c%f", tsd);
```

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:

```
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 £7.99Total with staff discount is £6.791500
```

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

YES. £7.99 - discount(15%) =

100% = 7.99

10% = 0.799

1% = 0.0799

7.99 - 1.985 = 6.79

ANSWER should be £6.79

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

Screenshot(s) of output:

```
Enter product 1: pizza
.... and the price please: -2.54
Re-enter the price please: -2.34
Re-enter the price please: |
```

(will do this until a valid number is entered)

for all products.

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

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

Screenshot(s) of output:

```
Enter product 1: pizza
.... and the price please: -2.54
Re-enter the price please: -2.34
Re-enter the price please:
```

Result: Did the programme behave as expected ie syntactically and logically correct?
NO, will not allow you to skip a product.

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:

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

Void main ()
{
    char product[5][15];
    float price[5], total = 0, tsd;

    for (int i = 0; i < 5; i++)
    {
        printf("Enter product %d: ", i + 1);
        scanf("%s", &product[i]);
        printf(".... and the price please: ");
        scanf("%f", &price[i]);

        while (price[i] < 0) //this is so that we dont give money away. there can still be free products
        {
            printf("Re-enter the price please: ");
            scanf("%f", &price[i]);
        }
    }

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

    for (int j = 0; j < 5; j++)
    {
        printf("%s ", product[j]);
        printf("\x9c%.2f\n", price[j]);
        total = total + price[j];
    }

    printf("Total price is \x9c%.2f", total);

    tsd = total * 0.15;
    tsd = total - tsd;

    printf("Total with staff discount is \x9c%f", tsd);
}
```


FINAL GRADE

/20

GENERAL COMMENTS

Worksheet 3:

Coding:

Good answer and quite efficient. Consider what would happen if entering a letter into a price

9/10

Testing:

Could be a bit more thorough, see my comment above. You've got the main points though.

8/10

Total: 17/20

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

Kallem G Thomas - 2304300

by Kallem Thomas

Submission date: 09-Jan-2024 03:21PM (UTC+0000)

Submission ID: 221206401

File name: 49116_Kallem_Thomas_Kallem_G_Thomas_-_2304300_541134_859580068.docx (46.69K)

Word count: 685

Character count: 3069

WORKSHEET 4 – TESTING (Arrays)

1. Original code:

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

#include "Header.h"

void main()
{
    char choice;
    int low = 1, high = 5;
    int myrand[3];
    int spin;
    int score = 100;
    srand(time(NULL));

    printf("Would you like to spin?: ");
    scanf(" %c", &choice);

    while (choice == 'y' || choice == 'Y')
    {
        for (int i = 0; i < 3; i++)
        {
            myrand[i] = rand() % (high - low + 1) + low;
            printf("%d ", myrand[i]);
        }

        if (myrand[0] == myrand[1] && myrand[0] == myrand[2])
        {
            if (myrand[0] == 5 && myrand[1] == 5 && myrand[2] == 5)
            {
                printf("\nJACKPOTT!");
                score = score + 2000;
            }
            else
            {
                score = score + 20;
            }
        }
        else if (myrand[0] == 5 && myrand[1] == 5 || myrand[1] == 5 && myrand[2] == 5 ||
myrand[0] == 5 && myrand[2] == 5)
        {
            score = score + 100;
        }
        score = score - 5;
        printf("\nYour score is: %d\n", score);
        printf("\nWould you like to spin again?: ");
        scanf(" %c", &choice);
    }
    if (choice == 'n' || choice == 'N')
    {
        printf("\nSee you later!\n");
    }
}
```

```
else
{
    printf("\nYou have not entered a valid choice");
}

if (score <= 0)
{
    printf("You have run out of points, you lose!");
}
```

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:

```
Would you like to spin?: y
4 4 4
Your score is: 20

Would you like to spin again?: |
```

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

Test: ii. 1 3 2 (0 points)

Screenshot(s) of output:

```
Would you like to spin?: y
1 3 2
Your score is: 0

Would you like to spin again?: |
```

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:

vii. 1 5 5 (100 points)

Screenshot(s) of output:

```
Would you like to spin?: y
1 5 5
Your score is: 100

Would you like to spin again?: |
```

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

Test:

viii. 5 5 5 (2000 points)

Screenshot(s) of output:

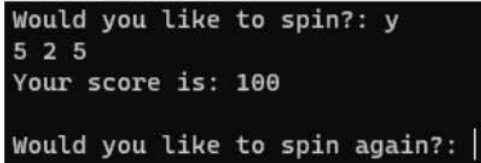
```
Would you like to spin?: y
5 5 5
JACKPOTTT!
Your score is: 2000

Would you like to spin again?: |
```

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

Test: 5 2 5

Screenshot(s) of output:



```
Would you like to spin?: y
5 2 5
Your score is: 100

Would you like to spin again?: |
```

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 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:

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

#include "Header.h"

void main()
{
    char choice;
    int low = 1, high = 5;
    int myrand[3];
    int spin;
    int score = 100;
    srand(time(NULL));

    printf("Would you like to spin?: ");
    scanf(" %c", &choice);

    while (choice == 'y' || choice == 'Y')
    {
        for (int i = 0; i < 3; i++)
        {
            myrand[i] = rand() % (high - low + 1) + low;
            printf("%d ", myrand[i]);
       }
    }
}
```

```

        if (myrand[0] == myrand[1] && myrand[0] == myrand[2])
        {
            if (myrand[0] == 5 && myrand[1] == 5 && myrand[2] == 5)
            {
                printf("\nJACKPOTT!");
                score = score + 2000;
            }
            else
            {
                score = score + 20;
            }
        }
        else if (myrand[0] == 5 && myrand[1] == 5 || myrand[1] == 5 && myrand[2] == 5 ||
myrand[0] == 5 && myrand[2] == 5)
        {
            score = score + 100;
        }
        score = score - 5;
        printf("\nYour score is: %d\n", score);
        printf("\nWould you like to spin again?: ");
        scanf("%c", &choice);
    }
    if (choice == 'n' || choice == 'N')
    {
        printf("\nSee you later!\n");
    }
    else
    {
        printf("\nYou have not entered a valid choice");
    }

    if (score <= 0)
    {
        printf("You have run out of points, you lose!");
    }
}

```

FINAL GRADE

/20

GENERAL COMMENTS

Worksheet 4:

Coding:

Good - you've implemented the cumulative score. A lot of the features were present from the initial code, well done. You could have rationalised the code a bit to make it more efficient. For example You have the values stored in an array, so you could have taken the option of using loops: particularly useful for counting 5s rather than long if statements.

8/10

Testing:

Would have liked to have seen what happens if you select "N" to spin again, otherwise good.

9/10

Total: 17/20

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

Kallem G Thomas - 2304300

by Kallem Thomas

Submission date: 09-Jan-2024 03:20PM (UTC+0000)

Submission ID: 220268586

File name: 49116_Kallem_Thomas_Kallem_G_Thomas_-_2304300_541134_509352285.docx (67.74K)

Word count: 738

Character count: 3866

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 a letter that isn't Y, y, n or N when the game asks if you would like to play.

Screenshot(s) of output:

```
That isn't a valid choice, please try again (Y/N): |
```

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

Test: input a letter when the game asks to choose a position between 0 and 15

Screenshot(s) of output:

```
Your choice was: 0

1 0 0 0
0 0 0 0
0 0 0 0
0 0 0 0
Missed, please try again.
Choose position to strike (0-15): |
```

Result: Did the programme behave as expected ie syntactically and logically correct?
YES, a letter in the system doesn't have a value, therefore a letter = 0.

Test: input -1

Screenshot(s) of output:

```
That's out of bounds, please try again.  
Choose position to strike (0-15): |
```

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

Test: input 17 (greater than the choice)

Screenshot(s) of output:

```
That's out of bounds, please try again.  
Choose position to strike (0-15): |
```

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

Test: does the system show what position has already been struck when missed?

Screenshot(s) of output:

```
Your choice was: 2  
  
1 1 1 0  
0 0 0 0  
0 0 0 0  
0 0 0 0  
Missed, please try again.  
Choose position to strike (0-15): |
```

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

Test: does the system show you where you've won on the map area when correct?

Screenshot(s) of output:

```
Your choice was: 4

1 1 1 1
8 0 0 0
0 0 0 0
0 0 0 0
You win!! Well done.
Would you like to play again? (Y/N): |
```

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

Test: if you play another game, does it re randomise the area? (Previous round above: won = 4)

Screenshot(s) of output:

```
Your choice was: 4

0 0 0 0
1 0 0 0
0 0 0 0
0 0 0 0
Missed, please try again.
Choose position to strike (0-15): |
```

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

Test: does your system stop if you say you would like to stop playing once a round has ended?

Screenshot(s) of output:

```
Understandable, have a nice day.
```

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

Test: does the game welcome you into the game each round?

Screenshot(s) of output:

```
-----  
Welcome to the treasure hunt game.  
-----  
Would you like to play? (Y/N): |
```

(Round 1)

```
-----  
Welcome to the treasure hunt game.  
-----  
Choose position to strike (0-15): |
```

(Round 2+)

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

Final code:

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

int Area[16] = { 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0 };

#include "Header.h"

void main()
{
    char number = -1;
    int choice;
    int win;
    int low = 1, high = 16;
    int myrand;
    char play;
    int count = 1;
    srand(time(NULL)); //seed randomness
    printf("-----\n\nWelcome to the treasure hunt game.\n\n-----\n\nWould you like to play? (Y/N): ");
    scanf("%c", &play);

    while (play != 'y' && play != 'Y' && play != 'n' && play != 'N')
    {
        system("cls");
        printf("That isn't a valid choice, please try again (Y/N): ");
        scanf("%c", &play);
    }

    while (play == 'y' || play == 'Y')
    {
        Welcome(count);
        myrand = rand() % (high - low + 1) + low;
        printf("Choose position to strike (0-15): ");
        scanf("%s", &number);

        choice = atoi(&number);

        while (choice < 0 || choice > 16)
        {
            printf("That's out of bounds, please try again.\n");
            printf("Choose position to strike (0-15): ");
            scanf("%s", &number);

            choice = atoi(&number);
        }

        while (choice >= 0 && choice <= 16)
        {
            system("cls");

            printf("\nYour choice was: %d\n", choice);
            win = table(choice, myrand);
```

```

while (choice != myrand)
{
    printf("Missed, please try again.\n");
    printf("Choose position to strike (0-15): ");
    scanf("%s", &number);

    choice = atoi(&number);
    system("cls");
    printf("\nYour choice was: %d\n", choice);

    while (choice < 0 || choice > 16)
    {
        system("cls");
        printf("That's out of bounds, please try again.\n");
        printf("Choose position to strike (0-15): ");
        scanf("%s", &number);

        choice = atoi(&number);
    }

    win = table(choice, myrand);
}

printf("You win!! Well done.");

printf("\nWould you like to play again? (Y/N): ");
scanf(" %c", &play);

while (play != 'y' && play != 'Y' && play != 'n' && play != 'N')
{
    system("cls");
    printf("That isn't a valid choice, please try again (Y/N): ");
    scanf(" %c", &play);
}

//The reset

for (int i = 0; i < 16; i++)
{
    Area[i] = { 0 };

}
count++;
choice = -1;
system("cls");
}

if (play == 'n' || play == 'N')
{
    printf("Understandable, have a nice day.");
}

```

```
}
```

HEADER FILE:

```
int table(int UserChoice, int Rando)
{
    int count = 0;

    char choice;
    int win;

    win = Area[Rando];
    if (UserChoice == Rando)
    {
        Area[UserChoice] = 8;

    }
    else
    {
        Area[UserChoice] = 1;

    }

    printf("\n");

    for (int i = 0; i < 16; i++)
    {
        printf("%d ", Area[i]);
        count++;
        if (count == 4)
        {
            printf("\n");
            count = count - 4;

        }

    }

    Area[UserChoice] = 1;
    return win;
}

int Welcome(int count)
{
    if (count >= 2)
    {
        printf("-----\n\nWelcome to the treasure hunt game.\n\n-----\n\n");
    }

    return 0;
}
```

FINAL GRADE

/30

GENERAL COMMENTS

Worksheet 5:

Coding:

Watch your validation - you ask the user to enter 0-15 but it does allow 16 to be entered.

Good use of functions and you're using the return correctly. It would be stylistically better to have the board display on its own I think (One function - one purpose) as it makes it more reusable (though I am nitpicking a bit!) Similarly it seems strange to have "Welcome to the treasure hunt game" in both main and a function - one copy would be better and then reuse it.

Might have been nice to extend the program to count the turns of the user, but overall a good effort.

11/15

Testing:

Generally good, though you have missed a few aspects (see my comment about the validation above - remember to test everything, including the boundaries of your validation, this is where erroneous values are most likely to get through.

10/15

Total: 21/30

Organisation of report, writing style, presentation of work:

Generally good, writing style is appropriate. A little terse at times, nothing wrong with that, but make sure you have explained everything you need to clearly even if you need to use a few more words.

8/10

Overall total for assignment: 80.5%

Test mark: 85%

Overall mark for module: 82%

PAGE 1
PAGE 2
PAGE 3
PAGE 4
PAGE 5
PAGE 6
PAGE 7