

Programs in C and Documentation

ACCA4017 Software Development

Ben Atkins 1809782

Table of Contents

- a) Introduction
- b) Part A
 - 1) What is a variable and why are they important to writing programs?
 - 2) Explain why the for-loop can also be referred to as the counting-loop and give an example of its use.
 - 3) Explain what an array is, what their features are and give an example of when you would use one.
- c) Part B
 - 1) Exercise 1
 - 2) Exercise 2
 - 3) Exercise 3
 - 4) Exercise 4
- d) Conclusion
- e) Reference List

A – Introduction

Throughout this assignment the author will discuss the firstly some of the principles within coding and C as well as how and when they are used and some examples of their use. It will then move onto some exercises on coding in C with the steps for implementation, annotated code and results from testing the programs the author has written. They will discuss what they have learned from these exercises and the thought process involved.

B - Part A – Understanding

A1 – What is a variable and why are they important to writing programs?

Variables are very important within programming, this is due to them enabling flexible programming. Instead of having to hard code data into programs for it perform calculations you may instead create an identifier to signify where the data will be input into the program and where that data will be manipulated and used[1]. This becomes particularly useful for programs that will be used routinely as in these circumstances different data will most likely have to be input to the program and for this purpose the result will also need to change dependent upon the data input, this is where programs will start to vary hence the name variable as these variable values will cause different effects dependant on their current assigned value instead of having to edit the code and manually enter the values needed. If there were no variables there would be no change in outcome from the program unless other, changing factors such as time were used.

A2 – Explain why the for-loop can also be referred to as the counting-loop and give an example of its use.

The for-loop can be known as the counting loop due to its condition within its syntax, it follows a process of initialisation, condition and increment. The first step initialisation is typically used to declare and initialise a variable for the conditional and increment step, the initialisation step does not need to be filled but is useful to use. The second step, condition, is then used to provide the

conditions that must be met to initialise the loop i.e. $X < 10$, This will mean that whilst 'X' is less than 10 the loop will run repeatedly. Lastly, is the increment this step is typically used to modify the variable that governs the conditions for the loop, but can be used for other purposes, similarly to the initialisation step this is not required for the loop to function[2]. For these reasons the for loop is dubbed the counting loop as it uses incremental conditions to govern whether the nested block of code will run and at the same time provide a measurable variable for how many times the code has been executed.

For-loops are typically implemented when a chunk of code is required to be repeated but it is known how many times it must be repeated[3]. An example of this use could be when you would like a user to input 12 values into an array for a calculation and could be displayed as so;

```
Void main()
{
    Int i;
    Float prices[13];
    For (i=0; i<12; i++)
    {
        Printf("please enter a value for item number %d: ", prices[i+1]);
        Scanf("%f", &prices[i]);
    }
}
```

A3 – Explain what an array is, what their features are and give an example of when you would use one.

An array is a collection of data grouped into one variable stored in a two-dimensional data matrix[4], this can be called upon then to denote whichever value stored you require as it stores values in numeric sequence with the following syntax; datatype variablename[x]; with x being the number of values you need to store in the array. An example of this can be seen below from a Learn C page [5];

```
// Program to find the average of n (n < 10) numbers using arrays
#include <stdio.h>
int main()
{
    int marks[10], i, n, sum = 0, average;
    printf("Enter n: ");
    scanf("%d", &n);
    for(i=0; i<n; ++i)
    {
        printf("Enter number%d: ", i+1);
        scanf("%d", &marks[i]);
    }
}
```

```

        sum += marks[i];
    }
    average = sum/n;
    printf("Average = %d", average);
    return 0;
}

```

In this example it is shown that the array 'marks' is used to store several integer variables, as shown by the 'int' prefix when it is declared, it also shows how a for loop is also used to add all of the values in the array together by using the variable 'i' within the variable's value field in order to add all integer's stored within the array into the sum variable. This is a very good example of using arrays as normally using just variables there would need to be several 'marks' variables declared in order to store them all which is tedious to write and code and will take up a lot of space on the page whereas using it this way is very line-efficient and will take far less time.

C – Part B – Implementation

1) Exercise 1

1.1 Identification of problem

- Users enter in 3 values, one their height in meters (float), one for their age (integer), and one for their name (character string)
- Store values in respective variables
- Print now stored variables on screen with appropriate prefixes and suffixes i.e. xx years old

1.2 Annotated Code

```

#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <time.h>

void main()
{
    //exercise 1
    float height; // initial declaration of variables and assigning it a value
    char name[99]; // must explicitly tell the number of characters for variable
plus one
    int age; //variable to hold users age as integer
    printf("\nPlease enter your height in meters: ");
    scanf("%f", &height);
    printf("\nPlease enter your name: ");
    scanf("%s", &name);
    printf("\nPlease enter a value for your age: ");
    scanf("%d", &age);
    printf("%.2f Meters \n%s \n%d Years old", height, name, age); //.2
    before 'f' is to state to display to 2 decimal places
}

```

1.3 Testing

1.3.1 Correct Values

Result: program ran fully and successfully and displayed correct values.

```
C:\Users\nwi_be208938\source\repos\Project4\Debug\Project4.exe
Please enter your height in meters: 1.9
Please enter your name: Ben
Please enter a value for your age: 19
1.90 Meters
Ben
19 Years old
```

1.3.2 Incorrect Values – wrong data types

Result: parts of the code were skipped and data was displayed incorrectly, putting incorrect data types into age caused program to indefinitely loop printing text and was unable to be captured.

```
Please enter your height in meters: edcrvtb
Please enter your name:
Please enter a value for your age: 29
-107374176.00 Meters
edcrvtb
29 Years old
```

1.3.3 Lower Values

Result: program ran through correctly and fully and displayed low data values correctly

```
C:\Users\nwi_be208938\source\repos\Project4\Debug\Project4.exe
Please enter your height in meters: 0.1
Please enter your name: a
Please enter a value for your age: 0
0.10 Meters
a
0 Years old
```

1.3.4 High Values

Result: program ran through fully, incorrect data was displayed for numeric fields however.

```
C:\Users\nwi_be208938\source\repos\Project4\Debug\Project4.exe
Please enter your height in meters: 9999999999
Please enter your name: vnldfsivbelsvutrliqberligburlgounglbivubefslvidbrglfuvnligbbihrgbvlgdibk
Please enter a value for your age: 9999999999999
17865419145578191840608256.00 Meters
vnldfsivbelsvutrliqberligburlgounglbivubefslvidbrglfuvnligbbihrgbvlgdibk
1316134911 Years old
```

2. Exercise 2

2.1 – Identification of the problem

- User enters in 3 numeric values for the radius of a circle, the base of a triangle and the height of a triangle
- 3 values are stored in appropriately named float variables
- validate that the triangle is not larger than the circle
- pass triangle height and base length to a function to calculate area of a triangle using $(B \times H) / 2$
- return area
- Store returned value in a float
- Pass the circle's radius to a function to calculate area of a circle using πr^2
- return area
- store returned value in a float
- subtract area of a triangle from the area of circle
- print value on screen in cm^2

2.2 – Annotated Code

```
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <time.h>
#define Pi 3.14
float CalcCircle(float);
float CalcTri(float, float);
float CalcShade(float, float);

void main()
{
    //exercise 2
    float TriBase = 0, TriHeight=0, CircRad, CircArea, TriArea, total; //majority
    declared without initial values, other are assigned values to allow loops to function
    accordingly in first instance
    printf("\nPlease enter a value for the radius of the circle: ");
    scanf("%f", &CircRad);
    while (TriBase > ((CircRad * 2) - 2) || TriHeight > ((CircRad * 2) - 3) || TriBase
    < 1 || TriHeight < 1) //Loop is to prevent triangle becoming larger than the circle
    {
        printf("\nPlease enter a value for the Triangle's Base: "); //these are to
        determine the values for the measurements of the circle and triangle
        scanf("%f", &TriBase);
        printf("\nPlease enter a value for the Triangle's Height: ");
        scanf("%f", &TriHeight);
        if (TriBase > ((CircRad * 2) - 2) || TriHeight > ((CircRad * 2) - 3) || TriBase
        < 1 || TriHeight < 1)
        {
            printf("Values for triangle were Too large for the dimensions of the circle
            entered, please re-enter them");
        }
    }
}
```

```

}
CircArea=CalcCircle(CircRad); //passes circle radius to a function to calculate
TriArea = CalcTri(TriBase, TriHeight);
total = CalcShade(CircArea, TriArea);
printf("\nthe Area of the Shaded section was %.2fcm^2", total);
}

float CalcCircle(float Radius) //used in exercise 2
{
    float Area; //temporary variable used to store and return
    Area = ((pow(Radius, 2))*Pi); //calculates area using known formula
    return Area; //returns this value to main to be stored in CircArea
}

float CalcTri(float Base, float height)
{
    float Area;
    Area = (Base * height) / 2;
    return Area;
}

float CalcShade(float CircArea, float TriArea)
{
    float Area;
    Area = CircArea - TriArea;
    return Area;
}

```

2.3 – Testing

2.3.1 – Correct Values

```
C:\Users\nwi_be208938\source\repos\Project5\Debug\Project5.exe

Please enter a value for the radius of the circle: 12

Please enter a value for the Triangle's Base: 5

Please enter a value for the Traignle's Height: 7

the Area of the Shaded section was 434.66cm^2
```

Result: program ran through fully and displayed the correct values

2.3.2 – Incorrect values, wrong data types

[illegible]

Result: upon entering letters into the circle radius field an endless loop occurred whereby it was impossible to enter data, the code fully broke

2.3.3 – low values

```
Please enter a value for the radius of the circle: 1
Please enter a value for the Triangle's Base: 1
Please enter a value for the Traignle's Height: 1
Values for triangle were Too large for the dimensions of the circle entered, please re-enter them
Please enter a value for the Triangle's Base: 1
Please enter a value for the Traignle's Height: 0.5
Values for triangle were Too large for the dimensions of the circle entered, please re-enter them
Please enter a value for the Triangle's Base: 0.5
Please enter a value for the Traignle's Height: 0.5
Values for triangle were Too large for the dimensions of the circle entered, please re-enter them
Please enter a value for the Triangle's Base: 0.1
Please enter a value for the Traignle's Height: 0.1
Values for triangle were Too large for the dimensions of the circle entered, please re-enter them
Please enter a value for the Triangle's Base: -1
Please enter a value for the Traignle's Height: -1
Values for triangle were Too large for the dimensions of the circle entered, please re-enter them
Please enter a value for the Triangle's Base: -2
Please enter a value for the Traignle's Height: -2
Values for triangle were Too large for the dimensions of the circle entered, please re-enter them
Please enter a value for the Triangle's Base: -2
Please enter a value for the Traignle's Height: Values for triangle were Too large for the dimensions of the circle entered, please re-enter them
Please enter a value for the Triangle's Base: -2
Please enter a value for the Traignle's Height: Values for triangle were Too large for the dimensions of the circle entered, please re-enter them
Please enter a value for the Triangle's Base:
```

Result: program would not allow me to enter very low data or negative numbers

2.3.4 – High Values

```
C:\Users\nwi_be208938\source\repos\Project5\Debug\Project5.exe
Please enter a value for the radius of the circle: 84645634856345
Please enter a value for the Triangle's Base: 2395645
Please enter a value for the Traignle's Height: 23975629
the Area of the Shaded section was 22497734074892860804487446528.00cm^2
```

Result: program ran through fully and outputted a value for the area of the shaded section

3. Exercise 3

3.1 – Identification of the problem

- allow users to enter 5 prices in by looping a function 5 times
- store these values in a float array
- Use another loop to add all the values within the array together
- print the current total on the scree
- Multiply the total value by 1.2 to get the total after VAT
- print the VAT value on the scr

3.2 – Annotated code

```
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <time.h>

float EnterPrice(int);

Void main()

{
```

```

//Exercise 3
int i;
float prices[5], total1 = 0;
for (i = 0; i < 5; i++) //first for loop to get the user to input all prices at
once
{
    prices[i] = EnterPrice(i);
}
for (i = 0; i < 5; i++) //second for loop to get the total of all prices
{
    total1 = total1 + prices[i];
}
printf("\nThe Total of all the prices was %.2f \nWith VAT added (20%) it will
be %.2f\n\n", total1, total1*1.2);
}

float EnterPrice(int Value)
{
    float price = 0;
    printf("\nPlease Enter a price for item number %d: ", Value + 1); //variable +1
being displayed for usability
    scanf("%f", &price);
    return price;
}

```

3.3 – Testing

3.3.1 – Correct Values

```

Please Enter a price for item number 1: 5
Please Enter a price for item number 2: 2
Please Enter a price for item number 3: 8
Please Enter a price for item number 4: 5
Please Enter a price for item number 5: 2

The Total of all the prices was 22.00
With VAT added (20) it will be 26.40

```

Result: program ran through fully and displayed the correct values

3.3.2 – Incorrect data types

```

Please Enter a price for item number 1: d
Please Enter a price for item number 2:
Please Enter a price for item number 3:
Please Enter a price for item number 4:
Please Enter a price for item number 5:
The Total of all the prices was 0.00
With VAT added (20) it will be 0.00

```

Result: upon entering 1 wrong data type the program completely skipped through and did not allow me to enter in any further data

3.3.3 – negative values

```
C:\Users\nwi_be208938\source\repos\Project5\Debug\Project5.exe
Please Enter a price for item number 1: -2
Please Enter a price for item number 2: -89
Please Enter a price for item number 3: -38.40
Please Enter a price for item number 4: -22.65
Please Enter a price for item number 5: -32.6
The Total of all the prices was -184.65
With VAT added (20) it will be -221.58
```

Result: the program ran through correctly and displayed the total correctly

3.3.4 – all values the same

```
Please Enter a price for item number 1: 5
Please Enter a price for item number 2: 5
Please Enter a price for item number 3: 5
Please Enter a price for item number 4: 5
Please Enter a price for item number 5: 5
The Total of all the prices was 25.00
With VAT added (20) it will be 30.00
```

Result: the correct values are calculated and displayed correctly

4. Exercise 4

4.1 - Identification of the problem

- Code runs to spin function
- generate a random number between 1 and 5
- return randomly generated number
- Store number within an integer array
- repeat 2 more times for a total of 3 randomly generated numbers using a for loop
- check the results using the checker function
- If the numbers meet the point prize criteria display the appropriate message and add the number of points to the number of points won
- prompt the user with a y/n prompt for if they want to continue
- If the response is 'yes' use a while loop to repeat the 3 spin and checker function

- if the response is 'no', end the program

4.2 – Annotated code

```
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <time.h>
int spin();
int checker(int, int, int, int);
void main()
{
    //Exercise 4
    srand(time(NULL)); //seed random, uses sys clock, only needed once... ever in a
    program
    int result[4], a, points = 0;
    char ch = 'y';
    for (a = 0; a < 3; a++) // for loop to provide 3 values for the integers within
    the array
    {
        result[a] = spin(); //initial spin
    }
    printf("%d %d %d", result[0], result[1], result[2]);
    points = checker(result[0], result[1], result[2], points); //initial call of
    checker funtion to verify the results of the first spin
    //winning results go here
    while (ch == 'y')
    {
        printf("\nWould you like to spin again? [Y/N]: ");
        scanf(" %c", &ch);
        if (ch == 'y' || ch == 'Y')
        {
            for (a = 0; a < 3; a++) //nested for loop to allow user to repeat
            code many times
            {
                result[a] = spin();
            }
            printf("%d %d %d", result[0], result[1], result[2]);
            points = checker(result[0], result[1], result[2], points);
        }
        else if (ch == 'n' || ch == 'N') //allow user to exit program by
        inputting 'n'
        {
            exit;
        }
        else //any incorrect entrie allows user to re-enter and informs them of
        their mistake
        {
            printf("\nplease re-enter you decision\n");
            ch = 'y'; //choice variable set to 'y' to allow the while loop to
            repeat
        }
    }
}
int spin()
{
    int num = 0;
    int low = 1, high = 5; //range for random numbers
```

```

        num = rand() % (high - low + 1) + low; //generates random number to variable %=
        modulus
        return num;
    }
    int checker(int num1, int num2, int num3, int points)
    {
        if (num1 == 5 && num2 == 5 && num3 == 5) //if statement chain to filter out the
        relevant winning results
        {
            points = points + 2000;
            printf("\nJackpot! +2000 points\nyour current point count is %d",
points);
        }
        else if (num1 == 5 && num2 == 5)
        {
            points = points + 100;
            printf("\nWell done, 2 fives +100 points\nyour current point count is
%d", points);
        }
        else if (num1 == 5 && num3 == 5)
        {
            points = points + 100;
            printf("\nWell done, 2 fives +100 points\nyour current point count is
%d", points);
        }
        else if (num3 == 5 && num2 == 5)
        {
            points = points + 100;
            printf("\nWell done! 2 fives +100 points\nyour current point count is
%d", points);
        }
        else if (num1 == num2 && num2 == num3)
        {
            points = points + 20;
            printf("\nGood Job! 3 of a kind +20 points\nyour current point count is
%d", points);
        }
        else
        {
            printf("\nBetter luck next time");
        }
        return points;
    }
}

```

4.3 Testing

4.3.1 – Correct Values

```

1 1 1
Good Job! 3 of a kind +20 points
your current point count is 520
Would you like to spin again? [Y/N]: y
5 5 1
Well done, 2 fives +100 points
your current point count is 100
Would you like to spin again? [Y/N]:
2 5 5
Well done! 2 fives +100 points
your current point count is 200
Would you like to spin again? [Y/N]: y
5 5 5
Jackpot! +2000 points
your current point count is 3380
Would you like to spin again? [Y/N]: y

```

```

4 4 1
Better luck next time
Would you like to spin again? [Y/N]: y
3 3 4
Better luck next time
Would you like to spin again? [Y/N]: y
1 3 4
Better luck next time
Would you like to spin again? [Y/N]: y

```

Result: when using correct response all prizes can be acquired and points system works correctly

4.3.2 – incorrect values

```
Would you like to spin again? [Y/N]: q
please re-enter you decision
Would you like to spin again? [Y/N]: h
please re-enter you decision
Would you like to spin again? [Y/N]: 3
please re-enter you decision
Would you like to spin again? [Y/N]: [
please re-enter you decision
Would you like to spin again? [Y/N]:
```

Result: entering incorrect values notifies users that they have entered an incorrect value and allows them to re-enter their decision

4.3.3 – duplicate information (2 'y's or 'n's)

```
Would you like to spin again? [Y/N]: yy
3 4 1
Better luck next time
Would you like to spin again? [Y/N]: 2 3 5
Better luck next time
Would you like to spin again? [Y/N]: yyy
4 4 3
Better luck next time
Would you like to spin again? [Y/N]: 2 5 1
Better luck next time
Would you like to spin again? [Y/N]: 1 2 2
Better luck next time
Would you like to spin again? [Y/N]:
```

Result: entering in 2 'y's causes code to repeat strangely and run through 2 spin functions, using 'n' twice will cause the code to immediately exit

d) Conclusion

In conclusion the author has learned throughout this report about the importance of arrays, variables and for-loops in programming as well as some of their more applied and specific application for use. As well as this by completing the coding exercises they have developed their skills of integrating several functions in C together and using all of the skills they have been taught in order to construct resilient programs. The author also feels that they have developed good skills with testing and that these tasks have provided them with knowledge to carry out testing on larger more complex applications.

e) Reference List

- [1] {NTC}, "Information on how the Variables work and what they are being used for," {*NTC*}. [Online]. Available: <https://www.ntchosting.com/encyclopedia/scripting-and-programming/variable/>. [Accessed: 25-Feb-2019].
- [2] Tutorials Point, "for loop in C," *Tutorials Point*. [Online]. Available: https://www.tutorialspoint.com/cprogramming/c_for_loop.htm. [Accessed: 25-Feb-2019].
- [3] Jim's Computer Science, "Programming - For Loop." [Online]. Available: https://www.cs.utah.edu/~germain/PPS/Topics/for_loops.html. [Accessed: 25-Feb-2019].
- [4] Bharat Kinariwala & Tep Dobry, "Online EE 160 Textbook," *University of Hawaii*. [Online]. Available: <http://ee.hawaii.edu/~tep/EE160/Book/>. [Accessed: 25-Feb-2019].
- [5] Learn C, "C Arrays: Declare, Initialize and Access Elements (With Examples)," *Programiz*. [Online]. Available: <https://www.programiz.com/c-programming/c-arrays#example>. [Accessed: 25-Feb-2019].