

C Programming by K. Jennings

by Kai Bentley - jennings

FILE	11982_KAI_BENTLEY_- _JENNINGS_C_PROGRAMMING_BY_K._JENNINGS_173935_1650031956. DOCX (147.67K)		
TIME SUBMITTED	06-MAY-2019 04:18AM (UTC+0100)	WORD COUNT	2500
SUBMISSION ID	106376691	CHARACTER COUNT	15198

Programs in C and Documentation

Written by Kai Hakkeem Bentley – Jennings

Student Number: 1807048

Date: 05/05/2019

Contents Page

Part A:

A1 – A3 – Page 3

Part B:

Short Notice – Page 4

Exercise 1 – Pages 4 to 5

Exercise 2 – Page 6

Exercise 3 – Pages 7 to 8

Exercise 4 – Pages 9 to 11

The Entire Code for Testing:

Pages 12 - 14

Part A

A1: Variables unlike constants possess interchangeable data. A constant when referring to something like an integer is set in stone and remains the value it has been given in the program, for example a fixed number of 9 or such. A variable integer is designed in a way that allows the user to input data as it runs instead of needing to constantly change internal data, allowing multiple flexibility options when designing a program. A simple example would be entering your age, or the radius of a circle. This would be achieved by using the 'scanf' function to prompt the user for data. Good if you intend to recycle content.

A2: A for-loop is also known as a counting loop as it repeats until a certain value is reached. For each time the loop plays, it cumulatively increases. Using an integer and initializing its value in the loop, you can make it add 1 for each time the loop completes effectively causing it to 'count'. This is useful for something like multiple variables as an array, as it keeps the code required minimal and time saving as an alternate to repeating the same 2 lines for an additional 10 lines.

A3: An array consists of multiple constants and/or variables. It's convenient for housing data for an overall subject area such as temperatures, prices, or even for putting in full names instead of singular characters. An array is useful for calculating an average in a more streamlined fashion rather than adding several integers, and as such is good alongside a for-loop when asking the user for lots of information.

Part B

Short Notice:

Before I preview the code for my exercises, I will take this opportunity to just briefly show the libraries I included, as well as a define function and additional 'void mains' that I used. Please take care using the code as all exercises are designed to work one after the other and may require separating if this may bother you which I apologize for. I will paste the FULL code at the very end of this assignment.

```
#include <stdio.h> //library
#include <math.h>
#include <stdlib.h>
#include <time.h> //Randumb
#define PI 3.14

void AreaBot5000(void); // Ex2: Area removal
void MoneyBot5000(void); // Ex3: Product total
void GambleBot5000(void); // Ex4: Gamble the kids' college fund away
```

Exercise 1:

A variable can contain different types of values either each time restarting a program or even throughout the program where you intend to use that type of integer, float etc again. Below is the code I used for this exercise:

```
void main()
{
    //Exercise 1

    int age = 21;
    float height = 1.72;
    char myname[20] = { 'K', 'a', 'i' };
    int choice;

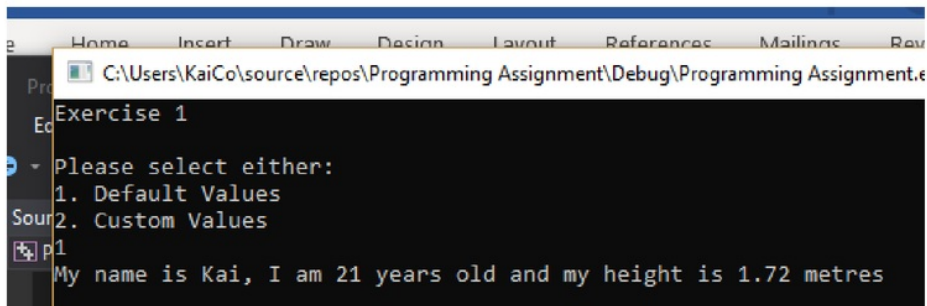
    printf("Exercise 1\n\n");
    printf("Please select either:\n1. Default Values\n2. Custom Values\n");
    scanf("%d", &choice);

    switch (choice)
    {
        case 1:
            printf("My name is %s, I am %d years old and my height is %.2f\n", myname, age, height);
            break;
        case 2:
            printf("Please enter your first name\n");
            scanf("%s", &myname);
            printf("Please enter your age\n");
            scanf("%d", &age);
            printf("Please enter your height in metres (i.e. 1.81)\n");
            scanf("%f", &height);

            printf("So your name is %s, you are %d years old and your height is %.2f metres", myname, age, height);
            break;
    }
    AreaBot5000();
    MoneyBot5000();
    GambleBot5000();
}
```

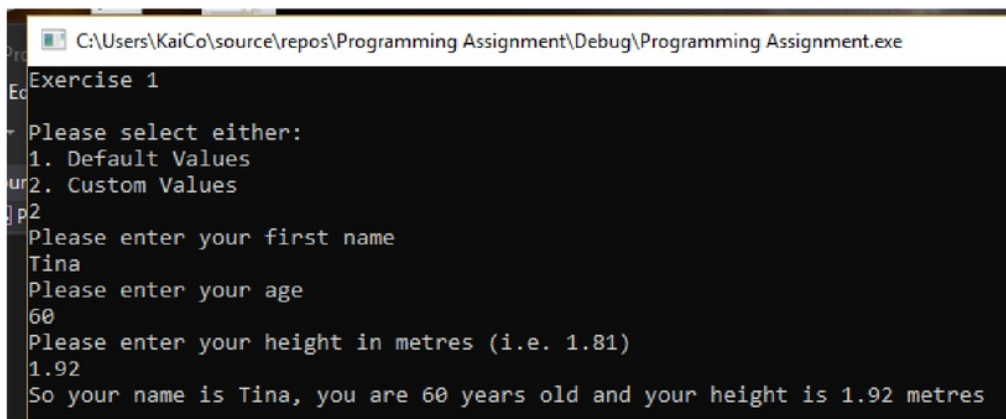
Kai Hakkeem Bentley – Jennings 1807048

As you can see, I have used constants for this exercise to demonstrate using my details. The program asks the user first if they would like to use the default values (my details) or to input custom ones. Below are mine when default is selected:

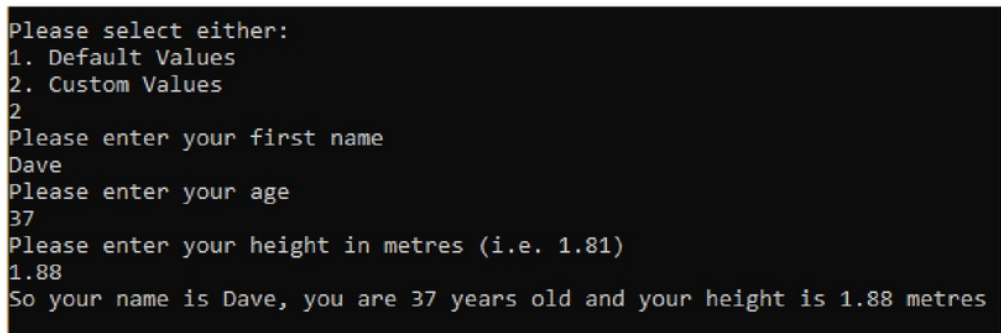


```
C:\Users\KaiCo\source\repos\Programming Assignment\Debug\Programming Assignment.exe
Exercise 1
Please select either:
1. Default Values
2. Custom Values
p1
My name is Kai, I am 21 years old and my height is 1.72 metres
```

This prints out onto the screen my constant values, whereas a custom result would look like these instead:



```
C:\Users\KaiCo\source\repos\Programming Assignment\Debug\Programming Assignment.exe
Exercise 1
Please select either:
1. Default Values
ur2. Custom Values
p2
Please enter your first name
Tina
Please enter your age
60
Please enter your height in metres (i.e. 1.81)
1.92
So your name is Tina, you are 60 years old and your height is 1.92 metres
```



```
Please select either:
1. Default Values
2. Custom Values
2
Please enter your first name
Dave
Please enter your age
37
Please enter your height in metres (i.e. 1.81)
1.88
So your name is Dave, you are 37 years old and your height is 1.88 metres
```

This overwrites the constants of my information and responds back to the user with their own. Variables are incredibly useful for tasks like this where it's necessary to know information that is likely to change.

Exercise 2:

For calculating the area of a formerly whole circle, this code was used to get the results needed:

```
void AreaBot5000()
{
    //Exercise 2

    float TriWidth, TriHeight;
    float CircRadius;
    float TriTotal;
    float CircTotal;
    float AreaTotal;

    printf("\n\nPlease enter the radius of your Circle\n");
    scanf("%f", &CircRadius);

    printf("Now please enter the width of your triangle\n");
    scanf("%f", &TriWidth);
    printf("Followed by the Height (From base to tip)\n");
    scanf("%f", &TriHeight);

    printf("Now...\n");

    CircTotal = PI * CircRadius * CircRadius;
    TriTotal = (TriWidth * TriHeight) / 2;
    AreaTotal = CircTotal - TriTotal;

    printf("Your original circle area was %.3f, and after taking away the
triangle's area of %.3f your circle now has an area of %.3f remaining!", CircTotal,
TriTotal, AreaTotal);
}
```

The program makes use of three different total areas, the triangular area, circular and remaining area after accounting the loss of the triangular area from the circle.

The process asks the user to input the radius of the circle, which calculated the area by multiplying it by itself, followed by multiplying it by Pi (which was shown earlier in the notice defined as 3.14). It will then ask the user for the height and width of the triangle, multiplying both and then dividing the result by two. Once these two totals are obtained, it will subtract the triangular area from the circular one, finally showing the remaining area shown below:

```
Please enter the radius of your Circle
10
Now please enter the width of your triangle
9
Followed by the Height (From base to tip)
9
Now...
Your original circle area was 314.000, and after taking away the triangle's area of 40.500 your circle now has an area of 273.500 remaining!
```

Exercise 3:

For this exercise I used an array to hold variable values for my product prices using a counting loop, as it would be more convenient than several float values to hold a single value. The counting loop also works well when using arrays, as the array can count alongside it, for example using an array to hold five values would be product[5], and you would use it in the loop defined as product[i] (i being a basic integer that is initialized in the loop and increased per loop) and stopping at the number before i.e. product[4] otherwise it will cause an overflow. My code below shows my method:

```
void MoneyBot5000()
{
    //Exercise 3

    float product[5];
    int i = 0;
    float VAT = 1.2;
    float PriceTotal = 0, RealPriceTotal = 0;

    printf("\n\nPlease enter the first product price please\n");
    scanf("%f", &product[i]);

    for (i = 1; i <= 4; i++)
    {
        printf("Please enter the next product price please\n");
        scanf("%f", &product[i]);
    }

    printf("Just confirming what you put in...\n");

    for (i = 0; i <= 4; i++)
    {
        printf("%.2f\n", product[i]);
    }

    printf("The total amount before VAT for all 5 products is... ");
    for (i = 0; i <= 4; i++)
    {
        PriceTotal = PriceTotal + product[i];
    }
    printf("\x9c%.2f\n", PriceTotal);
    printf("And after adding 20 percent VAT on top of your total brings the actual amount to... ");
    RealPriceTotal = PriceTotal * VAT;
    printf("\x9c%.2f!", RealPriceTotal);
}
```

I used the 20% increase for VAT as a float at 1.2 so that it is a little simpler for me to add the VAT to the total, as well as including an additional for-loop to add each value from the array to the total. The array is made to be a float in the event either the user or the end total generated contains a decimal, as even though I have used whole numbers in the screenshot below the total has come out with a decimal.

Kai Hakkeem Bentley – Jennings 1807048

```
Please enter the first product price please
4
Please enter the next product price please
6
Please enter the next product price please
5
Please enter the next product price please
5
Please enter the next product price please
8
Just confirming what you put in...
4.00
6.00
5.00
5.00
8.00
The total amount before VAT for all 5 products is... £28.00
And after adding 20 percent VAT on top of your total brings the actual amount to... £33.60!
```

Exercise 4:

This final exercise I decided to slightly change some factors. I have made it so that the machine awards the correct amounts requested by the paper (Triples that aren't fives award 20, two 5s 100 and triple fives 2000), but every time the user plays, they lose £5. It wouldn't be a slot machine without some risk, right?

As a result, instead of displaying £100 as a total when winning with two fives, it will instead display £95 as a *profit*. The program even tells you if you are making a loss as it subtracts 5 even on losses. It complicates things a little, but I thought it would be appropriate to include for a slot machine. I wouldn't have exactly earned 100 if I invested 5.

```
void GambleBot5000()
{
    int SlotReels[3];
    int low = 1, high = 5; //My number range
    srand(time(NULL)); //Seed
    int i, total = 0, bet = 5, reward1 = 20, reward2 = 100, jackpot = 2000;
    char betting; // (Y/N)`

    printf("\n\nBets are \x9c%d per spin. \n\nWould you like to play? (Y/N)\n",
bet);
    scanf(" %c", &betting);

    while(betting == 'y' || betting == 'Y')
    {
        total = total - bet; //Every play takes away 5 from the total

        for (i = 0; i <= 2; i++)
        {
            SlotReels[i] = rand() % (high - low + 1) + low;
            printf("%d ", SlotReels[i]);
        }

        if (SlotReels[0] == SlotReels[1] && SlotReels[1] ==
SlotReels[2] && SlotReels[2] != 5) //Triple numbers that isn't 5
        {
            printf("\nYou won \x9c%d!\n", reward1);
            total = total + reward1; // +15
        }
        else if (SlotReels[0] == SlotReels[1] && SlotReels[1] ==
SlotReels[2] && SlotReels[2] == 5) //Triple 5s
        {
            printf("\nCongratulations! You've won the Jackpot
of \x9c%d!!\n", jackpot);
            total = total + jackpot; // +995
        }
        else if (SlotReels[0] == 5 && SlotReels[1] == 5 &&
SlotReels[2] != 5 || SlotReels[0] == 5 && SlotReels[2] == 5 && SlotReels[1] != 5 ||
SlotReels[2] == 5 && SlotReels[1] == 5 && SlotReels[0] != 5) //Two 5s: 55*, 5*5, *55
        {
            printf("\nLucky you, you've earned \x9c%d!\n",
reward2);
            total = total + reward2; // +95
        }
        else
        {
            printf("\nSorry, looks like you don't win
anything\n");
        }

        if (total > 0)
```

```
        {
            printf("Your profit stands at \x9c%d\n", total);
        }
        else if (total < 0)
        {
            printf("You are currently at a loss of \x9c%d\n",
total);
        }
        else
        {
            printf("You are currently even at \x9c%d\n",
total);
        }

        printf("Would you like to play again? (Y/N)\n");
        scanf(" %c", &betting);
    }
}
```

It was a lot more convenient for me to write each reel to equal the other, rather than making several if statements for each individual number on triples. It was simple enough to make all of them equal the other, whilst making only one not equal to five. This is of course reversed for the triple fives, where all reels equal the other and one of them must equal five.

As far as I am aware, I think I needed to include the three variants of double fives as I didn't know any alternatives.

Upon meeting the conditions of one of the if statements, it will either award the player the specified amount of points as listed on the integer section or simply state they lost. 5 is always subtracted at the start of the spin rather than included into the total calculation. It was necessary to make 5 an integer as when using this '`\x9c`' along with the 5 right next to it, either became an error or a completely different symbol. That specific set of characters is to create the £ sign when using printf.

```
C:\Users\KaiCo\source\repos\Programming Assignment\
Bets are £5 per spin.

Would you like to play? (Y/N)
y
2 5 2
Sorry, looks like you don't win anything
You are currently at a loss of £-5
Would you like to play again? (Y/N)
y
1 5 2
Sorry, looks like you don't win anything
You are currently at a loss of £-10
Would you like to play again? (Y/N)
y
1 5 2
Sorry, looks like you don't win anything
You are currently at a loss of £-15
Would you like to play again? (Y/N)
y
5 4 4
Sorry, looks like you don't win anything
You are currently at a loss of £-20
Would you like to play again? (Y/N)
y
3 2 5
Sorry, looks like you don't win anything
You are currently at a loss of £-25
Would you like to play again? (Y/N)
y
```

An unfortunate run of bad luck trying to present my winnings that reached a hundred...

```
1 4 5
Sorry, looks like you don't win anything
You are currently at a loss of £-95
Would you like to play again? (Y/N)
y
5 5 1
Lucky you, you've earned £100!
You are currently even at £0
Would you like to play again? (Y/N)
y
1 2 1
Sorry, looks like you don't win anything
You are currently at a loss of £-5
Would you like to play again? (Y/N)
y
```

Finally won some money, also featuring the program taking into account the starting balance

```
4 1 1
Sorry, looks like you don't win anything
You are currently at a loss of £-75
Would you like to play again? (Y/N)
y
5 5 5
Congratulations! You've won the Jackpot of £2000!!
Your profit stands at £1920
Would you like to play again? (Y/N)
y
```

Despite a bad run I won the jackpot

The Entire Code for Testing

```
#include <stdio.h> //library
#include <math.h>
#include <stdlib.h>
#include <time.h> //Randumb
#define PI 3.14

void AreaBot5000(void); // Ex2: Area removal
void MoneyBot5000(void); // Ex3: Product total
void GambleBot5000(void); // Ex4: Gamble the kids' college fund away

void main()
{
    //Exercise 1

    int age = 21;
    float height = 1.72;
    char myname[20] = { 'K', 'a', 'i' };
    int choice;

    printf("Exercise 1\n\n");
    printf("Please select either:\n1. Default Values\n2. Custom Values\n");
    scanf("%d", &choice);

    switch (choice)
    {
    case 1:
        printf("My name is %s, I am %d years old and my height is %.2f\n", myname, age, height);
        break;
    case 2:
        printf("Please enter your first name\n");
        scanf("%s", &myname);
        printf("Please enter your age\n");
        scanf("%d", &age);
        printf("Please enter your height in metres (i.e. 1.81)\n");
        scanf("%f", &height);

        printf("So your name is %s, you are %d years old and your height is %.2f metres", myname, age, height);
        break;
    }
    AreaBot5000();
    MoneyBot5000();
    GambleBot5000();
}

void AreaBot5000()
{
    //Exercise 2

    float TriWidth, TriHeight;
    float CircRadius;
    float TriTotal;
    float CircTotal;
    float AreaTotal;

    printf("\n\nPlease enter the radius of your Circle\n");
    scanf("%f", &CircRadius);

    printf("Now please enter the width of your triangle\n");
```

```
scanf("%f", &TriWidth);
printf("Followed by the Height (From base to tip)\n");
scanf("%f", &TriHeight);

printf("Now...\n");

CircTotal = PI * CircRadius * CircRadius;
TriTotal = (TriWidth * TriHeight) / 2;
AreaTotal = CircTotal - TriTotal;

printf("Your original circle area was %.3f, and after taking away the
triangle's area of %.3f your circle now has an area of %.3f remaining!", CircTotal,
TriTotal, AreaTotal);
}

void MoneyBot5000()
{
    //Exercise 3

    float product[5];
    int i = 0;
    float VAT = 1.2;
    float PriceTotal = 0, RealPriceTotal = 0;

    printf("\n\nPlease enter the first product price please\n");
    scanf("%f", &product[i]);

    for (i = 1; i <= 4; i++)
    {
        printf("Please enter the next product price please\n");
        scanf("%f", &product[i]);
    }

    printf("Just confirming what you put in...\n");

    for (i = 0; i <= 4; i++)
    {
        printf("%.2f\n", product[i]);
    }

    printf("The total amount before VAT for all 5 products is... ");
    for (i = 0; i <= 4; i++)
    {
        PriceTotal = PriceTotal + product[i];
    }
    printf("\n\n%.2f\n", PriceTotal);
    printf("And after adding 20 percent VAT on top of your total brings the actual
amount to... ");
    RealPriceTotal = PriceTotal * VAT;
    printf("\n\n%.2f!", RealPriceTotal);
}

void GambleBot5000()
{
    int SlotReels[3];
    int low = 1, high = 5; //My number range
    srand(time(NULL)); //Seed
    int i, total = 0, bet = 5, reward1 = 20, reward2 = 100, jackpot = 2000;
    char betting; // (Y/N)`

    printf("\n\nBets are \n\n%.2f per spin. \n\nWould you like to play? (Y/N)\n",
bet);
```

```
scanf(" %c", &betting);

while(betting == 'y' || betting == 'Y')
{
    total = total - bet; //Every play takes away 5 from the total

    for (i = 0; i <= 2; i++)
    {
        SlotReels[i] = rand() % (high - low + 1) + low;
        printf("%d ", SlotReels[i]);
    }

    if (SlotReels[0] == SlotReels[1] && SlotReels[1] ==
SlotReels[2] && SlotReels[2] != 5) //Triple numbers that isn't 5
    {
        printf("\nYou won \x9c%d!\n", reward1);
        total = total + reward1; // +15
    }
    else if (SlotReels[0] == SlotReels[1] && SlotReels[1] ==
SlotReels[2] && SlotReels[2] == 5) //Triple 5s
    {
        printf("\nCongratulations! You've won the Jackpot
of \x9c%d!!\n", jackpot);
        total = total + jackpot; // +995
    }
    else if (SlotReels[0] == 5 && SlotReels[1] == 5 &&
SlotReels[2] != 5 || SlotReels[0] == 5 && SlotReels[2] == 5 && SlotReels[1] != 5 ||
SlotReels[2] == 5 && SlotReels[1] == 5 && SlotReels[0] != 5) //Two 5s: 55*, 5*5, *55
    {
        printf("\nLucky you, you've earned \x9c%d!\n",
reward2);
        total = total + reward2; // +95
    }
    else
    {
        printf("\nSorry, looks like you don't win
anything\n");
    }

    if (total > 0)
    {
        printf("Your profit stands at \x9c%d\n", total);
    }
    else if (total < 0)
    {
        printf("You are currently at a loss of \x9c%d\n",
total);
    }
    else
    {
        printf("You are currently even at \x9c%d\n",
total);
    }

    printf("Would you like to play again? (Y/N)\n");
    scanf(" %c", &betting);
}
}
```

C Programming by K. Jennings

GRADEMARK REPORT

FINAL GRADE

63/100

GENERAL COMMENTS

Instructor

Part A

Variables A bit short - you need to talk about the different types of variable eg int, float, char and some naming conventions for variables too 5/10

Counting Loops You haven't quite stated the point that a counted loop knows the number of iterations before the loop starts. You have sort of hinted at this though! You need a code example 5 /10

Arrays - You need to state that arrays hold multiple values *of the same type* which again you hint at but don't quite say. You cannot have an array of constants. You need to give a coded example. 4.5 /10

Part B

Exercise 1

Code-Annotation Good overall, but don't forget to include validation - age in this context should be 0-120 and height in the context of human height should have limits too. 3.5/5

Documentation - Be careful when talking about constants - these are a specific thing in C. I think you mean "pre-set values" rather than "constant values" You need to present a breakdown of logic stages eg 1 Prompt user to enter name, 2 Prompt

user to enter age etc. testing is OK, but don't forget to test for negative values 2.5/5

Exercise 2

Code-Annotation - A working program has been produced, but again, think about validation. radius width etc should not be negative and the circle dimensions also need to be sensible in relation to the triangle values (ie the triangle shouldn't be bigger than the circle 6/10

Documentation - Some breakdown shown, just make this into numbered stages. Testing is a bit limited - show the full range of inputs eg negative numbers, triangle bigger than the circle etc 2.5/5

Exercise 3

Code-Annotation - Array use is good and a working program produced. Don't forget to validate for negative prices 8/10

Documentation - A reasonable breakdown (number stages as stated before) Testing is again quite limited to just the "happy path" - use the full range of possible user inputs (including negative numbers) 3/5

Exercise 4

Code-Annotation Good implementation and good use of the array. Some of the conditions on your else ifs are redundant but overall, a working program with accumulator. Well done. 17/20

Documentation - Same comment as before on breakdown. You haven't tested all the eventualities on the testing - you need to do this to make sure that all the if statements work correctly. 6.5/10

Total: 63.5%

PAGE 1
PAGE 2
PAGE 3
PAGE 4
PAGE 5
PAGE 6
PAGE 7
PAGE 8
PAGE 9
PAGE 10
PAGE 11
PAGE 12
PAGE 13
PAGE 14