

C programming Review

Lecture 2

Introduction

- Variable
- Conditional statements
- Iteration (while, for....)

Declare Variables

- A variable is a location in memory that is most frequently referred to by its *name*, it can store a *value* of particular data_type and it has *an address and can also be referred to by its address*.
- Variable must be declared before they can be use; the format is
 - data_type variable_name; e.g. int var1

Format specifiers for printf.

- There are quite a number of format specifiers for printf. Here are the basic ones :
- %d print an int argument in decimal
- %ld print a long int argument in decimal
- %c print a character
- %s print a string
- %f print a float or double argument
- %p print the value of a pointer in its hex format
- %u print an unsigned integer (positive numbers)
-

Input/ output

- There are numerous ways to **input** data via the keyboard:
 - `scanf("%d", &input)` (**explain this code**)
 - `fgets(ArrayName, Array_Size, stdin)`: **gets** can cause problems
- There are numerous ways to output data (to screen):
 - `printf(" the value of the input is: %d", input);`
 - `putchar(input);`
 - `puts(input);`
- It is important to remember that you write a program two sets of people the **end user** and a **programmer**:
 - For an end user: "appropriate user/interface; e.g. prompt/output...." ;
 - For the programmer what is the code doing?

Conditional Statements

- A conditional statement?
 - **Relational expressions**
 - Has two parts: Operands and operator
 - `5 > 3; 5==3; 5 == 5;`
 - Returns a Boolean value: true/false
 - **Boolean operators**
 - Has two parts: Operands and operator
 - The three basic operators are: AND (`&&`); OR (`||`) and NOT (`!`)
 - Returns a Boolean Value

Conditional Expressions

Basic Types of conditional statements:

- **If (condition 1){**
 - Statement 1....
- **}**
- Statement 2

- **If (condition 2)**
- **{**
 - Statement 1
- **}**
- **else**
- **{**
 - Statement 2;
- **}**
- Statement 3;

- If condition 1 is true
what is the output?
- If condition 1 is false
what is the output?
- If condition 2 is true
what is the output?
- If condition 2 is true
what is the output?

Iteration statements

- What is the purpose of iteration statements?
- Main types of Iteration statements are:
 - *while ; do while* and *for* loop:
 - When should you use them?
 - Sentinel and counter control?
 - Differentiate between them.
- The **3** Important steps associated with iteration statements:
 - Initialise conditional variables,
 - write conditional expression ;
 - change a variable in the conditional expression

What is the purpose of the following code?

```
denis.manley@apollo: ~/OS2/week1

int main()
{
    // variable declaration
    int value;

    // do while loop executed at least once

    do
    {
        printf("enter an integer between 1 and 20: ");
        scanf("%d",&value);
    }while ((value < 1) || (value > 20));

    printf("exit loop: the value entered is: %d\n", value);

    return 0;
}
```

What is the purpose of the following code?

```
denis.manley@apollo: ~/OS2/week1

int main()
{
    // variable declaration
    int value;

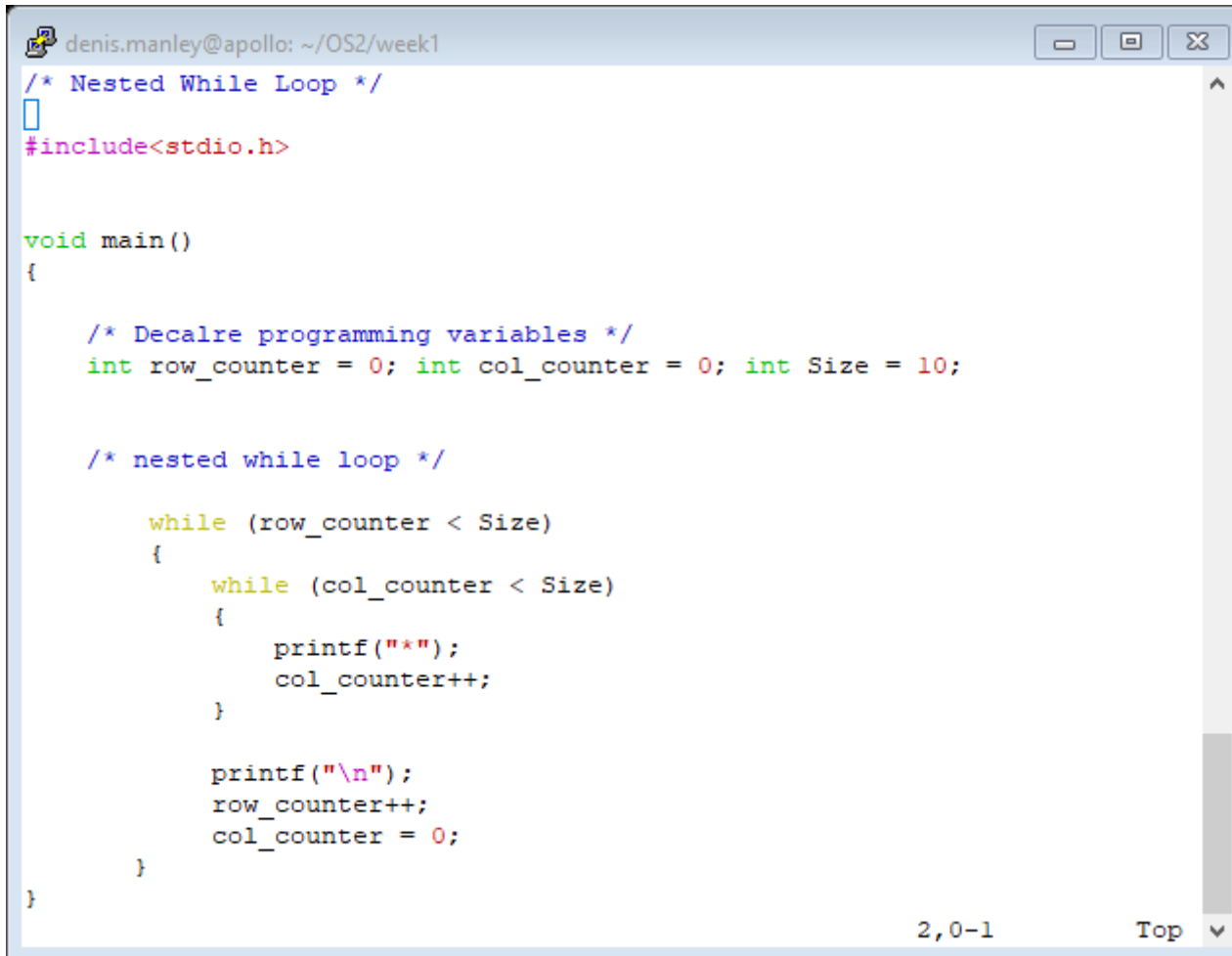
    // do while loop executed at least once

    do
    {
        printf("enter an integer between 1 and 20: ");
        scanf("%d",&value);
    }while ((value < 1) || (value > 20));

    printf("exit loop: the value entered is: %d\n", value);

    return 0;
}
```

a) What is the output of the following code?

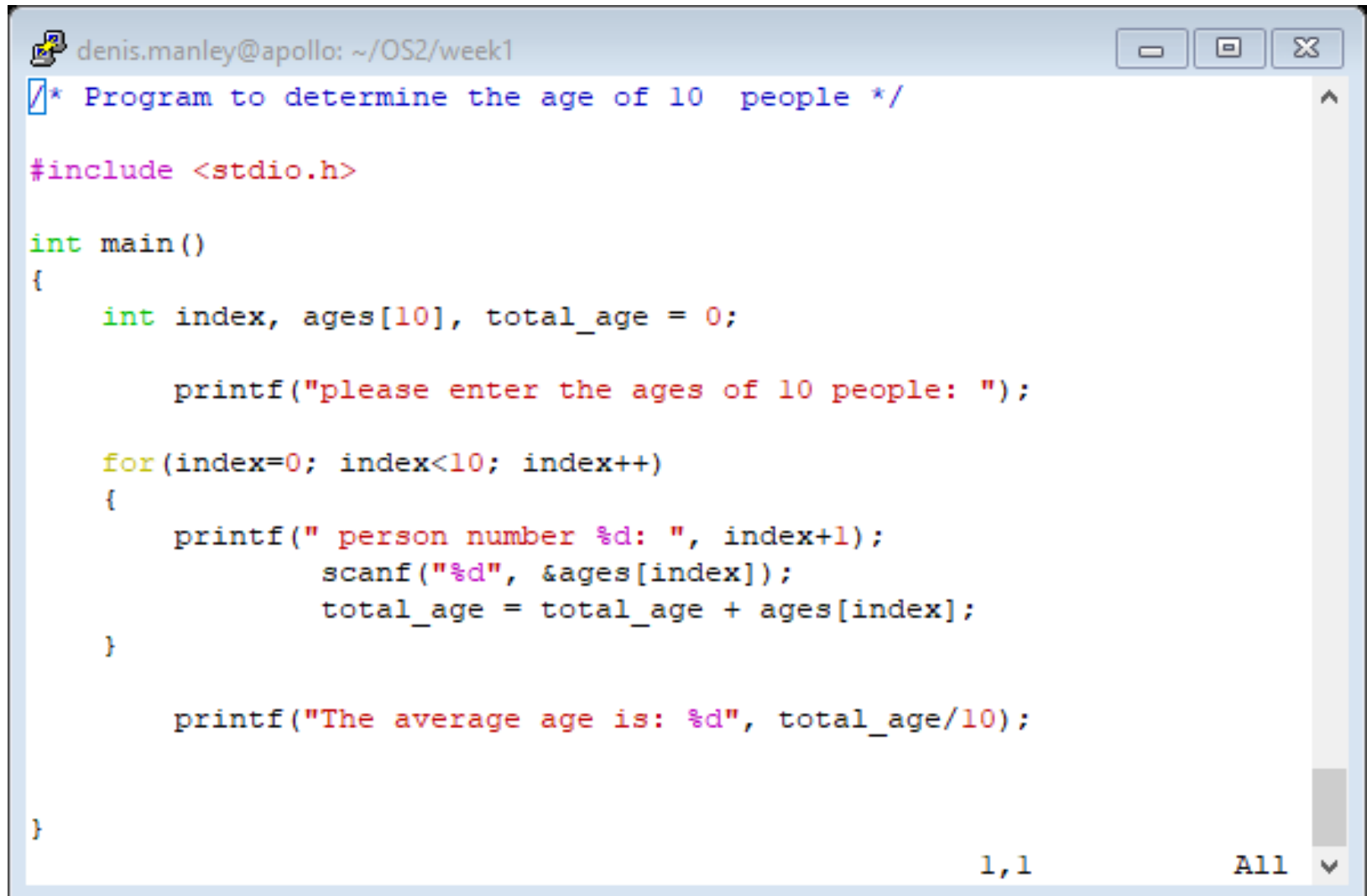
A screenshot of a terminal window with a light blue title bar. The title bar contains the text 'denis.manley@apollo: ~/OS2/week1' and three window control buttons (minimize, maximize, close). The terminal displays C code for a nested while loop. The code is as follows:

```
/* Nested While Loop */  
  
#include<stdio.h>  
  
void main()  
{  
  
    /* Decalre programming variables */  
    int row_counter = 0; int col_counter = 0; int Size = 10;  
  
    /* nested while loop */  
    while (row_counter < Size)  
    {  
        while (col_counter < Size)  
        {  
            printf("*");  
            col_counter++;  
        }  
  
        printf("\n");  
        row_counter++;  
        col_counter = 0;  
    }  
}
```

At the bottom right of the terminal window, the text '2,0-1' and 'Top' with a dropdown arrow are visible.

b) What would happen to the code if `col_counter = 0` is removed/commented out?

Arrays: static data structure



```
denis.manley@apollo: ~/OS2/week1
/* Program to determine the age of 10  people */

#include <stdio.h>

int main()
{
    int index, ages[10], total_age = 0;

    printf("please enter the ages of 10 people: ");

    for(index=0; index<10; index++)
    {
        printf(" person number %d: ", index+1);
        scanf("%d", &ages[index]);
        total_age = total_age + ages[index];
    }

    printf("The average age is: %d", total_age/10);

}
```

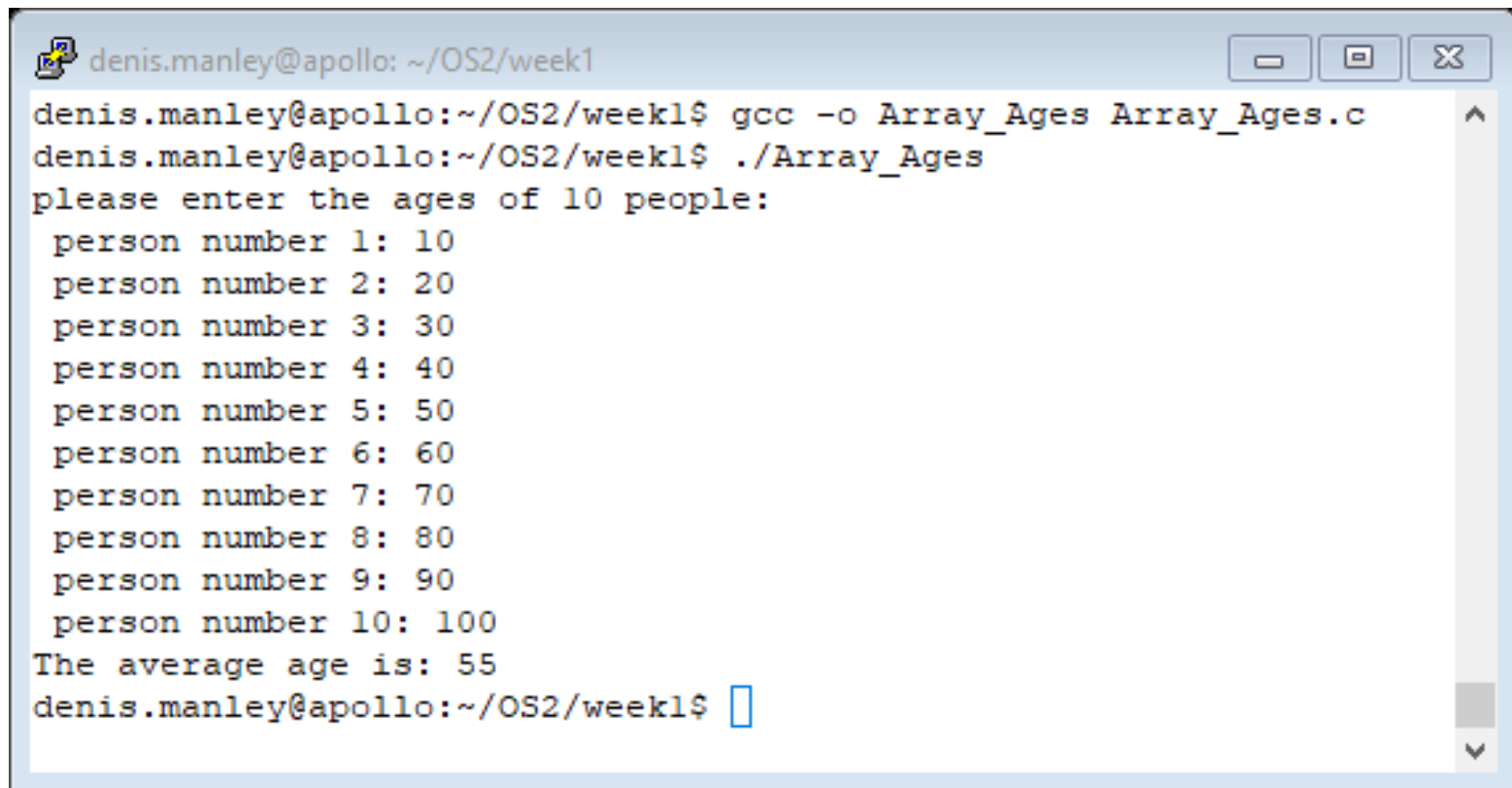
1,1 All

Compile and run a program in linux

- Use the **gcc compiler**
- To compile (go to directory where program is stored: using cd)
 - **gcc -o exe_file name c source file name**
- To run the program
 - **./exe file**

```
denis.manley@apollo:~/OS2/week1$ gcc -o DoWhile DoWhile.c
denis.manley@apollo:~/OS2/week1$ ./DoWhile
```

Array example: compile and run

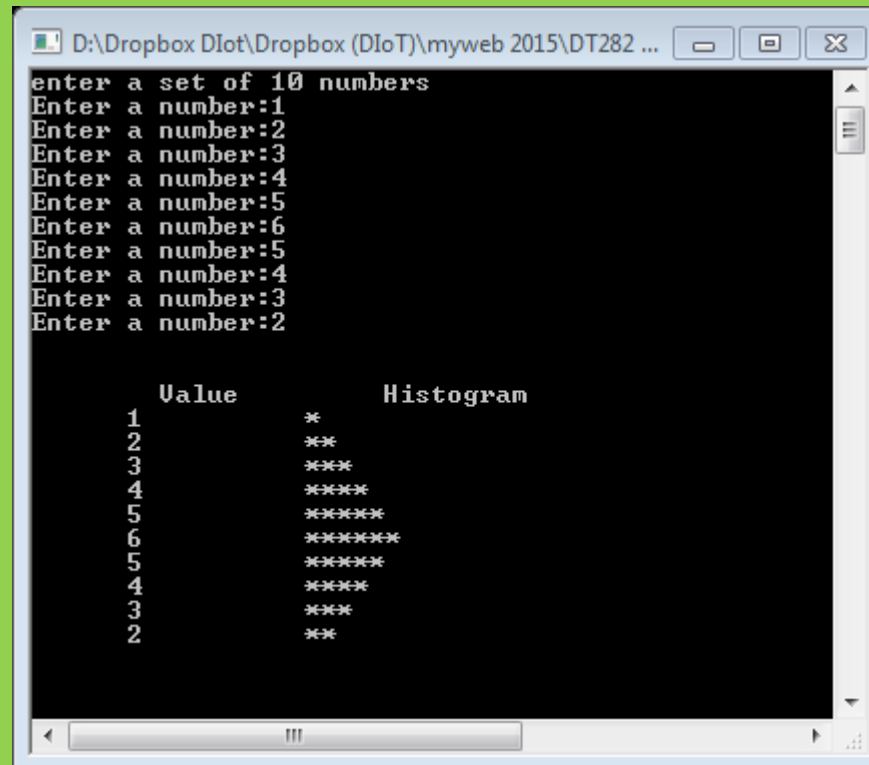


```
denis.manley@apollo: ~/OS2/week1
denis.manley@apollo:~/OS2/week1$ gcc -o Array_Ages Array_Ages.c
denis.manley@apollo:~/OS2/week1$ ./Array_Ages
please enter the ages of 10 people:
person number 1: 10
person number 2: 20
person number 3: 30
person number 4: 40
person number 5: 50
person number 6: 60
person number 7: 70
person number 8: 80
person number 9: 90
person number 10: 100
The average age is: 55
denis.manley@apollo:~/OS2/week1$
```

Does the program produce the correct output?

Week 1 Homework: Histogram

- Write an algorithm and a c program to
 - prompt the user for 10 numbers and store these in an array.
 - The numbers in the array are then processed to graph the information in the form of a bar chart or histogram - each number is printed, and then a bar consisting of that many asterisks is printed beside the number .
- The following is a sample of the expected output:



The screenshot shows a Windows command prompt window with the title "D:\Dropbox D\IoT\Dropbox (D\IoT)\myweb 2015\DT282 ...". The program prompts the user to "enter a set of 10 numbers" and then asks for each number individually. The input sequence is 1, 2, 3, 4, 5, 6, 5, 4, 3, 2. The output displays a histogram with two columns: "Value" and "Histogram". The "Value" column lists the numbers 1 through 2, and the "Histogram" column shows the corresponding number of asterisks for each value.

```
enter a set of 10 numbers
Enter a number:1
Enter a number:2
Enter a number:3
Enter a number:4
Enter a number:5
Enter a number:6
Enter a number:5
Enter a number:4
Enter a number:3
Enter a number:2

      Value      Histogram
      1          *
      2          **
      3          ***
      4          ****
      5          *****
      6          *****
      5          *****
      4          ****
      3          ***
      2          **
```