BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI (RAJ.)

Second Semester 2017-18   CS F111 Computer Programming

LABORATORY SESSION #5

(Types, Operators and Expressions)

1. Copy the file `/home/share/overflow.c` with the same name into your current directory, compile it and execute it. Take a look at the output carefully. Based on the output, what can you conclude about the internal representation of `char` datatype?

Next, declare `count` as an `unsigned char`:

```
  unsigned char count;  /* in place of char count */
```

Now compile and run the program. [Tip: An infinite loop is broken by pressing the <Ctrl> and C keys of the keyboard together]. What can you now conclude about the data representation of the variable `count`?

2. Explain why the following code prints the largest integral value on your system:

```
  unsigned long long val = −1;
  printf("The biggest integer value: %llu\n", val);
```

Note the conversion specifier **u** (for unsigned integers) preceded by the length modifier `ll` (ell-ell) used to print the value stored in val. Explore other length modifiers and conversion specifiers by reading the online manual for `printf()`, by typing the following command at the Linux shell prompt: `man 3 printf`. Write these in your lab notebook.

3. In this question, you will learn more about data types and qualifiers. Type out this program into a file named `lab5_sizes.c`. Use the *vi editor* and time saving commands (e.g., *dd* and *p* to copy and paste lines) to write this out.

```
#include<stdio.h>
int main()
{
      float f;
      printf("Sizeof (char) = %d bytes\n", sizeof(char));  // datatype
      printf("Sizeof (short)= %d bytes\n", sizeof(short));
      printf("Sizeof (int)= %d bytes\n", sizeof(int));
      printf("Sizeof (long)= %d bytes\n", sizeof(long));
      printf("Sizeof (float)= %d bytes\n", sizeof(f));  // variable
      printf("Sizeof (double)= %d bytes\n", sizeof(double));
      printf("Sizeof (1.55)= %d bytes\n", sizeof(1.55));  // constant
      printf("Sizeof (1.55L)= %d bytes\n", sizeof(1.55L));
      printf("Sizeof (str)= %d bytes\n", sizeof("Hello"));  // string
      return 0;
}
```

a. Find out the sizes of data types when prefixed with the keywords `signed` and `unsigned`.

b. Now, try various combinations of qualifiers (`short` and `long`) with the keywords `unsigned` and `signed` keywords. Try which of these the compiler accepts, and which are not. For example, `long unsigned int` is valid, whereas `long unsigned double` is not.

4. The following code is meant to give you practice with short-circuit evaluation:

```
  int a=0, b=0, x;
  x = 0 && (a=b=777);
  printf("%d %d %d\n", a, b, x);
  x = 777 || (a = ++b);
  printf("%d %d %d\n", a, b, x);
```

What gets printed? First, write your answers. Then write a test program to check them.

5. Search the Internet for ASCII table and glance through the it. Write a C program to print the characters corresponding to numbers 1 through 127 encoded by the ASCII scheme, one character per line, padded with ** and ** on either side of the character. For instance, the output corresponding to 65 should be printed this way:

```
65 corresponds to: **A**
```

Now modify your program to print characters that may correspond to numbers -128 to -1. What do you see?

6. As you know, the roots x1 and x2 of a quadratic equation $ax^2 + bx + c$ $(a = 0)$ are calculated by:

$$x1 = (-b + \sqrt{}\,(b^2 - 4ac))/2a,$$
$$x2 = (-b\ - \sqrt{}\,(b^2 - 4ac))/2a$$

Your program, named `lab5_quadroots.c` should take a, b and c as inputs, and output the values of x1 and x2. For calculating the square root, you can use the function called **`sqrt()`**, the declaration of which is in the header file `<math.h>` that you must #include, just as you do `<stdio.h>`. How to use this function in a C program? To find out the square root of x and store the result in y, you will write the following C code:

```
int x, y;
y = sqrt(x);
```

Using this information, write the entire program to output x1 and x2 (the roots).

Now try to compile the file `lab5_quadroots.c` as usual. Do you get an error message that says, “Undefined reference to sqrt”? Since we have a math feature in our program, we need to compile the program slightly differently. `$ gcc  lab5_quadroots.c  –lm` The option `–lm` helps in linking the math library to our program, myroots.

Test your program with the following test cases, and record the answers in your lab record:

i. a = 3, b = 5, c = 2
ii. a = 3, b = –6, c = 3
iii. a = 2, b = 1, c = 5

7. This code tries to calculate ‘one’ in three different ways: repeated adding, and two slight variants of multiplication. Try this and find out the output. Can you explain the results? (Note: A copy of this code is stored in the file `float_storage.c` in the directory `/home/share/`)

```
float f = 0.1f;
float sum = 0.0f, product;
int i;
for (i = 0; i < 10; ++i)
    sum += f;
product = f * 10;
printf("sum = %1.15f, mul = %1.15f, mul2 = %1.15f\n",
        sum, product, f * 10);
```

You may want to read this to get a better idea of what is happening: https://randomascii.wordpress.com/2012/02/25/comparing-floating-point-numbers-2012-edition/