ECM2433: The C family

C Workshop 1

Programming environment

The programs will be written, compiled/linked and run from the Linux command line. On a Blue Room machine log into Linux and start up the terminal window (from the menu: Applications; Accessories; Terminal). Alternatively, from a Windows machine use the Putty application to connect to one of the Linux room machines remotely (this is why you should not reboot or shutdown these computers).

While you may use any editor to write your programs, I strongly recommend that you become familiar with *vi*. It is not intuitive, but it comes as standard with all Linux and Unix installations and, in future, it may be the only option you have. There are many tutorials available online; try, for example,

http://heather.cs.ucdavis.edu/~matloff/UnixAndC/Editors/ViIntro.html

Getting started

At the command line, enter the following commands to create a new directory to hold your C programs:

```
cd pcfiles
mkdir ecm2412
cd ecm2412
```

The `pcfiles` directory is exactly the same as your U: drive under Windows. To make it easy to compile and your programs we will create a Linux shell script; create a file called `compileLink` with the following contents:

```
#!/bin/ksh

echo
echo Compiling ${1}.c
gcc -ansi -I./ -c ${1}.c -o ${1}.o

# Only do the link if the compilation returned no errors.

if [[ $? -eq 0 ]]
then
   echo Program compiled ok
   echo
   echo Linking
   gcc -L./ -lm ${1}.o -o ${1}

   if [[ $? -eq 0 ]]
   then
      echo Program compiled and linked ok
   else
      echo Link failed
   fi
else
   echo Compile failed
fi
```

To ensure that you are able to run this shell script, enter the command

```
chmod u+x compileLink
```

to give yourself execute permission. If you have written a program called `myProgram.c` that you wish to compile and link, enter the command

```
compileLink myProgram
```

Exercises

1. Hello World!

Write a C program called `hello.c` that prints out "Hello World!". Compile and link it by entering the command

```
compileLink hello
```

If this step is successful then you should find that your directory contains two new files: `hello.o` and `hello`. Run the program by typing

```
./hello
```

What happens if you remove the ***#include*** directive and then compile and link the program?

2. Variable declarations and printf

(a) Create a new program which defines three separate variables, assigns them the integer values 0, 100 and –123456 and prints out their values in the following format:

```
variable 1 is           0
variable 2 is         100
variable 3 is     -123456
```

(b) Define an array, populate it with the floating point values 0, 100.123 and -123456.7891 and print the values in a column, right-justified, to 2 decimal places, like this (with the trailing percentage sign):

```
      +0.00%
    +100.12%
-123456.79%
```

(c) Define three strings, use the ***strcpy*** function to populate them with the values "small", "middle-sized" and "absolutely enormously huge", and print out the values in a column which is exactly 19 characters wide, like this:

```
*tiny               *
*middle-sized       *
*absolutely enormous*
```

(d) Using the strings defined in (c), can you centre-justify the three strings? The output should look like this:

```
          tiny
      middle-sized
absolutely enormously huge
```

[Hint: you may wish to use the ***strlen*** function and the "%*.*s" formatting string to achieve this.]

3. Array bounds

Enter the following program (you will need to fill in some missing bits to make it a proper C program that will compile, link and run):

```
char  array2[5] = {'A','B','C','D','E'};
int   a[]       = {9, 8, 7, 6, 5, 4, 3, 2, 1, 0};
int   anArray[10];
int   a2[2][3]  = { {1, 2, 3} , {4, 5, 6} };

/* code to print out all the values in array a */
anArray[15] = 999;
/* code to print out all the values in array a */
```

What has happened to ***a*** and why?

Now remove the declarations for the ***array2***, ***a*** and ***a2*** variables (so that only ***anArray*** is remaining) and the parts of the program that print out the values in array ***a***. Compile and link the program (there should be no errors) and then run it. What happens and why?

Now remove the declaration for the anArray variable, leaving only the assignment statement. Compile/link the program. What happens and why?

4. Array of strings

Create a new program that stores three strings ("Hello World!", "test" and "Elephants") in a single array variable and then print out the three strings on separate lines.