Intro to C Programming

Summer 2011

Problem Set 1

Writing, compiling and debugging programs. Preprocessor macros. C file structure. Variables. Functions and program statements. Returning from functions.

Out: Thursday, June 8, 2011 **Due:** Thursday, June 15, 2011

Problem 1.1

(a) What do curly braces denote in C? Why does it make sense to use curly braces to surround the body of a function?

(b) Describe the difference between the literal values `7`, "7" and `'7'`.

(c) Consider the statement
```
double ans = 10.0 + 2.0/3.0 - 2.0*2.0;
```
Rewrite this statement, inserting parantheses to ensure that `ans = 10.0` upon evaluation.

Problem 1.2

Consider the statement

```
double ans = 18.0/squared(2+1);
```

For each of the four versions of the macro `squared()` below, write the corresponding value of `ans`.

1. `#define squared(x) x*x`
2. `#define squared(x) (x*x)`
3. `#define squared(x) (x)*(x)`
4. `#define squared(x) ((x)*(x))`

Problem 1.3

Write the "Hello, UNMLA students" program described in lecture in your favorite text editor and compile and execute it. Turn in a printout or screen shot showing:

- the command used to compile your program

- the command used to execute your program (using gdb)
- the output of your program

Problem 1.4

The following lines of code, when arranged in the proper sequence, output the simple message "All your base are belong to us."

1. `return 0;`
2. `const char msg[] = MSG1;`
3. }
4. `#define MSG1 "All your base are belong to us!"`
5. `int main(void) {`
6. `#include <stdio.h>`
7. `puts(msg);`

Write out the proper arrangement (line numbers are sufficient) of this code.

Problem 1.5

For each of the following statements, explain why it is not correct, and fix it.

(a)
```
#include <stdio.h>;
```

(b)
```
 int function(void arg1)
{
return arg1-1;
}
```

(c)
```
 #define MESSAGE = "Happy new year!"
puts(MESSAGE);
```

Problem Set
Types, operators, expressions

Problem 2.1

Determine the size, minimum and maximum value for the following data types. Please specify if your machine is 32 bit or 64 bits in the answer.

- char
- unsigned char
- short
- int
- unsigned long
- long long
- float

Hint: Use sizeof() operator, limits.h and float.h header files.

Problem 2.2
Write logical expressions that tests whether a given character variable `c` is:

- lower case letter
- upper case letter
- digit
- white space (includes space,tab,new line)

Problem 2.3

Consider `int val=0xCAFE;` Write expressions using bitwise operators that do the following:

- test if at least three of last four bits (LSB) are one
- reverse the byte order (i.e., produce `val=0xFECA`)
- rotate fourbits (i.e., produce `val=0xECAF`)

Problem 2.4

Using precedence rules, evaluate the following expressions and determine the value of the variables(without running the code). Also rewrite them using parenthesis to make the order explicit.

- Assume `(x=0xFF33,MASK=0xFF00)`. Expression: `c=x & MASK ==0;`
- Assume `(x=10,y=2,z=2;)`. Expression: `z=y=x++ + ++y2;`
- Assume `(x=10,y=4,z=1;)`. Expression: `y>>= x&0x2 && z;`

Problem 2.5

Determine if the following statements have any errors. If so, highlight them and explain why.

- `int 2nd_value=10;`
- Assume `(x=0,y=0,alliszero=1)`. `alliszero =(x=1) && (y=0);`
- Assume `(x=10,y=3,z=0;)`. `y=++x+y;z=z-->x;`
- Assume that we want to test if last four bits of `x` are one. `(int MASK=0xF;ison=x&MASK==MASK)`