

Problem Set 2 – Solutions

Types, operators, expressions

Out: Tuesday, January 12, 2010.

Due: Wednesday, January 13, 2010.

Problem 2.1

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

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

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

Answer: On my 32-bit machine (/usr/include/limits.h, /usr/include/float.h), the sizes and limits are as follows. Results may differ if you used a 64 bit machine.

Data type	size (bytes)	min	max
char	1	SCHAR_MIN (-128)	SCHAR_MAX (127)
unsigned char	1	0	UCHAR_MAX(255)
short	2	SHRT_MIN (-32768)	SHRT_MAX (32767)
int	4	INT_MIN (-2147483648)	INT_MAX (2147483647)
unsigned int	4	0	UINT_MAX(4294967295)
unsigned long	4	0	ULONG_MAX(4294967295)
float	4	FLT_MIN(1.175494e-38)	FLT_MAX(3.402823e+38)

Problem 2.2

Write logical expressions that tests whether a given character variable c is

- lower case letter (Answer: `c>='a' && c<='z'`)
- upper case letter (Answer: `c>='A' && c<='Z'`)
- digit (Answer: `c>='0' && c<='9'`)

- white space (includes space,tab,new line) (Answer: `c=='\n' || c=='\t' || c==' '`)

Problem 2.3

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

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

Answer:

- We have to test if last three or four bits are on. The possible values are `0x7,0xB,0xD,0xE,0xF`. To test this, first we extract the last four bits. (`int bits=val&0xF; /*last four bits*/`). Next we test if it is one of the possible patterns. (`bits==0x7 || bits==0xB || (bits>=0xD)`).
- `val = ((0xFF & val) << 8) | (val >> 8)`
- `val = (val >> 4) | ((val & 0xF) << 12)`

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++ + ++y*2;`
- Assume (`x=10,y=4,z=1`). Expression: `y>>= x&0x2 && z`

Answer:

- The operator precedence is `'=='>'&'>'`. Thus, the expression is equivalent to `c= (x & (MASK==0))`. Therefore `x=0xFF33,c=0`.
- The operator precedence is `'++>'*>'>'`. Thus, the expression is equivalent to `z=(x++) + ((++y)*2)`. Therefore `x=11,y=3,z=10+3*2=16`.
- The operator precedence is `'&&'>'&&'>'>=>'`. Thus, the expression is equivalent to `y>>= (x & 0x2) && z`. Therefore `x=10,y=2,z=1`.

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 on. (`int MASK=0xF;ison=x&MASK==MASK`)

Answer:

- (a) Variable names cannot start with a number.
- (b) '=' operator should be replaced with '=='. The correct version is `alliszero = (x==1) && (y==0);`.
- (c) There is nothing wrong with the statement. While `-->` may look suspicious, the expression simplifies to `y = (++x)+y; z=(z--)>x.`
- (c) There is nothing syntactically wrong with the statement. However, what we want is `ison=(x&MASK)==MASK.` Based on operator precedence, the current expression simplifies to `ison=x&(MASK==MASK)`

These exercises should have convinced you to use `()` always.

MIT OpenCourseWare
<http://ocw.mit.edu>

6.087 Practical Programming in C
January (IAP) 2010

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.