Homework 1---- Xiaohui Chen(xc2388)

2.57

```
#include <stdio.h>

typedef unsigned char *byte_pointer;

void show_bytes(byte_pointer start, int len) {
int i;
for (i = 0; i < len; i++)
printf(" %.2x", start[i]);
printf("\n");
}

void show_short(short x){
	show_bytes((byte_pointer)&x, sizeof(short));
}

void show_long(long x){
	show_bytes((byte_pointer)&x, sizeof(long));
}

void show_double(double x){
	show_bytes((byte_pointer)&x, sizeof(double));
}
```

2.59

```
(x & 0xFF) | (y & ~0xFF)
```

2.60

```
unsigned replace_byte (unsigned x, int i, unsigned char b)
{
	int tmp=sizeof(x);
	unsigned x3=(x<<8*(tmp-i))>>8*(tmp-i);
	unsigned x1=(x>>(i+1)*8)<<8;
	unsigned k=((x1+b)<<(8*i))+x3;
	return k;
}
```

2.61

A: !(~x)
B:!!˜x
C:!(~(x&0xFF))
D:!(x&0xFF000000)

2.63
unsigned srl(unsigned x, int k) {
/* Perform shift arithmetically */
unsigned xsra = (int) x >> k;
unsigned tmp=(~((!((1<<(sizeof(int)-1))&x))<<k))<<(sizeof(int)*8-k);
return tmp|xsra;
}

int sra(int x, int k) {
/* Perform shift logically */
int xsrl = (unsigned) x >> k;
unsigned tmp=~(tmp1<<(sizeof(int)-k));
return xsrl&tmp;
}

2.64
int any_odd_one(unsigned x)
{

   return x&0x55555555!=0;
}

2.66
int leftmost_one(unsigned x){
x=x-1;
x= x|(x>>1);
x= x|(x>>2);
x= x|(x>>4);
x= x|(x>>8);
x= x|(x>>16);
return x+1;
}

2.67
A: The shift of 32 bits in C is not defined. Expression x << k means shifit left by k mod 32.
B:
int bad_int_size_is_32() {
int tmp=0xFFFFFFFF;

```
return tmp+1==0;
}
C:
int bad_int_size_is_32() {
int set_msb=2<<15<<15;
int beyond_msb=set_msb<<1;
return set_msb && !beyond_msb;
}
```

2.68

```
int lower_one_mask(int n)
{
int tmp=1<<(n-1);
int k=tmp|(tmp-1);
return k;
}
```

2.69

```
unsigned rotate_left(unsigned x, int n)
{
int index=n/4;
unsigned tmp1=x>>(sizeof(x)-index);
unsigned tmp2=x<<index;
return tmp2|tmp1;
}
```

2.70

```
int fits_bits(int x, int n)
{
return !((~(x>>n))||(x>>n))
}
```