

## Question Set4

### *Bitwise operations*

#### 6.4

Given an unsigned char number a, copy it to another target, unsigned char b, in reverse order, so that a's LSB becomes b's MSB and vice versa.

For example:

```
a = 0111 0010
b = 0100 1110
```

#### 6.5b

Binary display.

Given some unsigned char a, implement a program which displays the number in a binary display.

For example:

```
int a = 125 = 0x7d ;
>>output: 111 1101
```

#### 6.16

Given two integers a, b, implement a function that returns max(a, b), using only bit operations and without > < operators.

Distinguish between cases: both negative, both positive, one is positive and the other negative.

### *Bit fields & Unions*

#### 11.1

You are given an unsigned short number: x .

Define the necessary auxiliary structures and unions, and compute the following:

- $\left\lfloor \frac{x}{256} \right\rfloor$
- $x \% 256$
- $x \% 16$

## Pointers to Functions

1.

Define 3 functions:

```
f1(x) = 7x+1; //linear
f2(x) = x^2 +2x +6; //square
f3(x) = x^3 +5x+ 1;//triple
```

- Implement a function that given 2 functions  $f, g$  and some value  $c$ , finds the  $\max(f(c), g(c))$ .
- Implement a function that receives an array of functions  $f_1...f_n$  and some value  $c$ , and returns the function  $f_i(c)$  which produces the maximum value:

*Syntax guide:*

```
typedef double(*func_t)(double);
func_t arr[N] ;
func_t MaxiamlFunc(func_t funcArr[N], double c);
//note: when returning a function's pointer, the
typedef is mandatory.
```

2.

- Define 3 comparing functions:

```
bigger(x, y) ; //checks who's bigger
bigger_abs(x, y); //checks who's bigger by absolute value
smaller(x, y); //checks who's smaller
```

- Implement a function `sortArr( )`, that receives an array of integers and some comparing function. `sortArr( )` will sort the array according to the sorting function.