

CSC112                      Spring 2011  
Fundamentals of Computer Science  
Lab 4  
Algorithms for Parts I and II

**Part I**

Write a C++ program that reads in two numbers, and stores those numbers as type “unsigned int”. Your program should then:

1. Print the sets corresponding to the two numbers. Your output should use set notation, e.g.,

`{ 'a', 'd', 'f' }`

2. Use a bit-wise logical operator to compute the set union of the two sets. Print your result.
3. Use a bit-wise logical operator to compute the set intersection of the two sets. Print your result.
4. Use a bit-wise logical operators to compute the set difference. Print your result. Note: Given two sets  $A$  and  $B$ , the set difference, denoted  $A - B$  is defined as all the elements that are in  $A$ , but not in  $B$ .

Test your program with input 53208257 and 8904049. You should get the sets:

`{ a g h k n o p q r t v y z }`    and    `{ a e f g i k l m o p q r s x }`

**Algorithm for Converting a Number to a Set**

Input:  $n$

```
current_letter = 'a'
while ( n > 0 )
    if ( the low order bit of n is 1 ) then
        output the current_letter
    end if
    increment the current_letter
    shift n one position to the right
end while
```

## Part II

Write a C++ program that reads a sequence of lower case letters (up to the end-of-line) and computes the numerical representation (unsigned int described above) of the set. Output the unsigned int as a number. Your program should check that the input sequence consists of ONLY lower case letters. If some other character other than 'a' through 'z' is found before the end of line, your program should print an error message and exit. Use your program from part I to verify that this part is working correctly.

### Algorithm Converting a Set of Letters to a Number

Input: A string of lower case letters 'a' to 'z'

```
u = 0 // u represents the input set as an integer
while ( we have not encountered a newline character )
    read the next letter
    if ( the letter just read is not a newline ) then
        convert the letter to a number k in the range 0 to 25
        set bit k of u to 1
    end if
end while
output u
```