

**CPSC 601: Programming with C & Java**  
**Spring 2009**  
**Homework 3**  
February 12, 2009

**Due date and time:** 3 p.m. Tuesday February 24, 2009

This set of homework consists of two Exercise problems from the textbook Chapter 5, and two from Chapter 7, totaling four problems.

Your program should have a head comment including problem number, a short description of the problem, and your name, and an appropriate amount of comments throughout the program. Also, your program should have a well thought-out logic and the program should be neatly coded.

- Chapter 5:
  - 5.4 [20 points] Writing a function `isvowel()` is required.
  - 5.6 [40 points]
- A modified version of 7.1 [30 points]:

Write a function called `previous_month()` that returns the previous month. Start with the code

```
enum month {jan = 1, feb, . . . . ., dec};  
typedef enum month month;
```

If `dec` is passed as an argument to the function, `nov` should be returned. If `jan` is passed as an argument to the function, `dec` should be returned.

Write another function that prints the name of a month. More explicitly, if the enumerator `jan` is passed as an argument, `January` should be printed.

Write `main()` so that it receives from the user an integer (in the range 1..12) and calls your functions and produces a list of six consecutive months starting with the previous month of the corresponding input integer. For example, if the user enters 3, the output should look like:

**February, March, April, May, June, July.**

For another example, if the user enters 1, the output should look like:

**December, January, February, March, April, May.**

- A modified version of 7.2 [30 points]:

Assume we have 28 days in February. Write a function called `next_day()`. The function should have two parameters to represent day and month. Use call-by-reference mechanism so that the two arguments can reflect the modified value of day and possibly month in case of crossing from one month to the next.

The program should take as input two integers – say, for example, 17 and 5, which represent 17 May, call the `next_day()` function to find the next day, and should print as output 18 May, which is the next day.

Use enumeration types. Pay particular attention to the problem of crossing from one month to the next.