CPSC 1375 — Programming I HW5

(Block letters via functions / procedures)

This exercise is designed to give you practice in input/output in C++, to introduce the data type char, and also to introduce you to using functions and procedures (that is, void functions) to modularize your code.

Firstly, write a function

```
void line (char ch , int num)
```

which outputs the character ch num times in a row on a given line. For example, line('A',10) should output:

```
AAAAAAAAAA
```

Secondly, write a function

```
void rectangle (char ch ,  int x ,  int y)
```

which writes the character ch in a rectangular pattern of x rows and y columns. For example, rectangle('X',3,5) should output:

```
XXXXX
XXXXX
XXXXX
```

[Note: Function "rectangle" should invoke function "line" to accomplish its job!!]

Thirdly, write a main program body to read one character at a time from the datafile "animals.dat", each record of which is the name of an animal. Your program should produce a rectangular pattern for each letter in the animal's name; the size of the rectangular pattern depends on the letter and on its position in the animal name, the rule being: if the x th letter in the name is the y th letter of the alphabet, the rectangle should be size x by y. For example, if the inputted animal name is CAT, your program should produce the output pattern:

```
CCC     ( 1 by 3, since the 1 st letter in CAT is the 3 rd letter
                  of the alphabet )
A
A       ( 2 by 1, since the 2 nd letter in CAT is the 1 st letter
                  of the alphabet)
TTTTTTTTTTTTTTTTTTTT
TTTTTTTTTTTTTTTTTTTT
TTTTTTTTTTTTTTTTTTTT  ( 3 by 20, since the 3 rd letter in CAT is
                        the 20 th letter of the alphabet)
```

(continued on back)

Note that you should skip one line between each block letter pattern, to separate them. You should also skip several lines between animal names, to separate each animal name from the next one.

Programming notes:

(1). Of course, your main program should call your procedure "rectangle" to produce the letter patterns!
(2). This program's structure is really a "single control break" one. You should read a character (while not EOF) and react appropriately depending on whether or not the character read is an end-of-record symbol '\n'. You will, of course, have to keep track of where in the particular animal name you are!
(3). Unlike most computer languages, C++ will let you do arithmetic with characters. This is particularly useful here, since you will want to somehow "convert" the character 'A' to the number 1, the character 'B' to the number 2, etc. (For simplicity, you can assume all letters in the data file are capital.)