16.216: ECE Application Programming

Summer 2015

Lecture 4: Key Questions

May 28, 2015

1. In what cases are `for` loops useful? Describe the basic structure of a `for` loop.

2. Describe the operators that allow you to directly modify a variable without writing a full assignment statement.

3. Explain the difference between pre- and post-increment or decrement operators.

4. **Example:** What does the following program print?

```
int n = 5;
printf("n = %d\n", ++n);
printf("Now, n = %d\n", n++);
printf("Finally, n = %d\n", n);
```

5. **Example:** What does each of the following print?

a.
```
for (i = 5; i < 40; i += 8)
{
    printf("%d ", i);
}
```

b.
```
for (i = -5; i < -10; i--)
{
  printf("%d ", i);
}
```

c.
```
for (i = 10; i <= 100; i = i+10)
{
    if (i % 20)
        printf("%d ", i);
}
```

d.
```
for (i = 5; i < 10; i += i%2)
{
  printf("%d ", i++);
}
```

In today's exercise, you will write a program that does the following:

- Prompts the user to enter a single input character followed by an integer, `n`.
    - If not correctly formatted, print error, clear rest of line, and repeat prompt
- Depending on the character entered, do the following:
    - '`F`' or '`f`': Compute and print the factorial of `n`, `n!`
        - For example, if the user enters **`F 5`**, print **`5! = 120`**
    - '`P`' or '`p`': Compute `2`^{n}, but only if `n >= 0`.
        - For example, if the user enters **`p 2`**, print **`2^2 = 4`**
        - Print an error if `n < 0`.
    - '`X`' or '`x`': Exit the program
    - In all other cases, print an error:
        - For example: `Invalid command Z entered`
- If the user enters any command other than '`X`' or '`x`', return to the initial prompt and repeat the program.

Steps in the programming exercise:

1. Draw a general flowchart for the overall program flow.
    - Treat each of the processes listed in part 2 as a single block—don't worry about the details just yet.
2. Draw smaller flowchart for reading the input character & integer until correct.
3. Draw smaller flowcharts for:
    - Computing n!
    - Computing 2^n if n >= 0 and printing an error otherwise.
4. Convert the flowcharts to actual code.

Space to draw flowchart/code for overall program flow:

Flowchart/code for reading input character until correct:

Flowchart/code for n!

Flowchart/code for 2^n