CSC / CIS 175

Problem Solving and Programming - I

University of Michigan-Flint
Department of Computer Science, Engineering, and Physics (CSEP)

Fall 2013

November 3, 2013

Homework 8

(100 points)

due by November 13, Wednesday 8:00am

Remarks:

- No emailed homeworks will be accepted.
- Only submission is via the BB system.
- No late submissions will be accepted.
- Individual assignment. No collaboration or group work is permitted.

Questions for the deliverable:

1. Write a program that simulates a coin tossing. For each toss of the coin, the program should print Heads or Tails. Let the program toss the coin 200 times and count the number of times each side of the coin appears. Print the results. The program should call a separate function (flip) that takes no arguments and returns 0 for tails and 1 for heads.

   Expected Output:

```
Tails Heads Heads Tails Heads Tails Heads Heads Tails Tails
Heads Heads Tails Heads Tails Tails Tails Heads Heads Tails
Tails Tails Tails Heads Tails Tails Heads Heads Tails Tails
Heads Heads Heads Tails Tails Heads Heads Tails Heads Tails
Tails Tails Tails Heads Heads Tails Tails Tails Heads Tails
Heads Heads Tails Tails Tails Heads Heads Heads Tails Tails
```

```
Tails Tails Tails Tails Heads Tails Tails Heads Heads Tails
Heads Tails Heads Tails Heads Tails Heads Tails Tails Heads
Tails Tails Heads Heads Heads Tails Tails Heads Tails Heads
Tails Heads Heads Tails Heads Heads Heads Heads Heads Heads
Tails Heads Heads Heads Heads Heads Tails Heads Tails Heads
Heads Heads Tails Tails Heads Heads Tails Tails Tails Tails
Tails Tails Tails Tails Tails Tails Heads Heads Heads Tails
Tails Heads Heads Heads Heads Tails Tails Heads Heads Heads
Tails Tails Heads Heads Heads Tails Tails Heads Tails Tails
Tails Heads Tails Heads Tails Tails Tails Tails Heads Heads
Heads Heads Heads Tails Tails Heads Tails Tails Heads Heads
Tails Tails Tails Tails Tails Heads Heads Heads Tails Tails
Heads Heads Heads Heads Heads Tails Tails Heads Tails Tails
Tails Heads Tails Tails Tails Heads Tails Tails Tails Tails

The total number of Heads was 94
The total number of Tails was 106
```

2. The expression $C(n, r)$ denotes the number of r-element subsets of an n-element set. For example, $C(4, 2)$ is 6 because there are six 2-element subsets of a 4-element set. The value of $C(n, r)$ is given by the formula

$$c(n, r) = \frac{n!}{r! * (n - r)!} \tag{1}$$

Write a program that computes $C(n, r)$ using the following component functions.

(a) `main`: prompts the user for two numbers and accents them into n and r, respectively.
(b) `check`: compares r and n. If $r > n$, check invokes the function `err_msg`, which prints an appropriate error message.
(c) `comb`: computes $C(n, r)$.
(d) `fact`: computes factorial.

Expected Output:

```
 (i)
Enter n in C(n,r) : 4
Enter r in C(n,r) : 7
 n must be greater than or equal to r in C(n,r)

  (ii)
Enter n in C(n,r) : 5
Enter r in C(n,r) : 2
10

 (iii)
```

```
Enter n in C(n,r) : 25
Enter r in C(n,r) : 20
53130

  (iV)
Enter n in C(n,r) : 52
Enter r in C(n,r) : 48
270725
```

3. Write a program to calculate the factorial by means of a function. Note: The result will not be accurate for relatively larger numbers with the built-in C++ data types. We will implement the accurate one later with arrays.

   Expected Output:

```
Expected Output:
  (i)
Enter a number for factorial computation and press ENTER: 5
5! (by Factorial Function) is  120

  (ii)
Enter a number for factorial computation and press ENTER: 23
23! (by Factorial Function) is  25852016738884978000000

   (iii)
Enter a number for factorial computation and press ENTER: 38
38! (by Factorial Function) is  523022617466601040000000000000000000000000000
```

4. Write a program that would ask user to input an integer and then print all the integers from 1 to that number in a function.

   Expected Output:

```
Enter a number (to print all whole numbers from 1 to that number) : 10
1 2 3 4 5 6 7 8 9 10
```

5. Write a program that will check and print if the numbers from 1 to 20 are prime or not as shown below. Implement checking the primeness in a function.

   Expected Output:

```
2 is prime
3 is prime
```

```
4 is not prime
5 is prime
6 is not prime
7 is prime
8 is not prime
9 is not prime
10 is not prime
11 is prime
12 is not prime
13 is prime
14 is not prime
15 is not prime
16 is not prime
17 is prime
18 is not prime
19 is prime
20 is not prime
```

6. Write a program that finds the first prime number greater than one billion.

   Expected Output:

   ```
   1000000007 is the first prime number after 1 billion
   ```

7. `Extra credit (8%)`: Write a program that prints a calendar for a year. Prompt the user for which day of the week January 1 is on and whether the year is a leap year. The day that January 1 is on is coded as follows: Sun 0 Mon 1 Tue 2 Wed 3 Thu 4 Fri 5 Sat 6 Hint: Use a switch statement inside a for or while loop for the monhts.

   Expected Output:

   ```
   Enter day of the week for January 1 4
   Enter day leap year code (1 for leap year, 0 for non-leap year) : 0


   January

   Su Mo Tu We Th Fr Sa
                 1  2  3
    4  5  6  7  8  9 10
   11 12 13 14 15 16 17
   18 19 20 21 22 23 24
   25 26 27 28 29 30 31
   ```

February

```
Su Mo Tu We Th Fr Sa
 1  2  3  4  5  6  7
 8  9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
```

March

```
Su Mo Tu We Th Fr Sa
 1  2  3  4  5  6  7
 8  9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31
```

April

```
Su Mo Tu We Th Fr Sa
          1  2  3  4
 5  6  7  8  9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30
```

May

```
Su Mo Tu We Th Fr Sa
                1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
31
```

June

```
Su Mo Tu We Th Fr Sa
    1  2  3  4  5  6
 7  8  9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
```

```
28 29 30

July

Su Mo Tu We Th Fr Sa
          1  2  3  4
 5  6  7  8  9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30 31

August

Su Mo Tu We Th Fr Sa
                   1
 2  3  4  5  6  7  8
 9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29
30 31

September

Su Mo Tu We Th Fr Sa
       1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30

October

Su Mo Tu We Th Fr Sa
             1  2  3
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30 31


November

Su Mo Tu We Th Fr Sa
 1  2  3  4  5  6  7
 8  9 10 11 12 13 14
```

```
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30

December

Su Mo Tu We Th Fr Sa
       1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31
```

8. `Extra credit (10%)`: Same as the above Extra credit question but output 3 months at a time as follows:

Expected Output:

```
Enter day of the week for January 1 4
Enter day leap year code (1 for leap year, 0 for non-leap year) : 0
```

```
    January                  February                 March

Su Mo Tu We Th Fr Sa     Su Mo Tu We Th Fr Sa     Su Mo Tu We Th Fr Sa
             1  2  3      1  2  3  4  5  6  7      1  2  3  4  5  6  7
 4  5  6  7  8  9 10      8  9 10 11 12 13 14      8  9 10 11 12 13 14
11 12 13 14 15 16 17     15 16 17 18 19 20 21     15 16 17 18 19 20 21
18 19 20 21 22 23 24     22 23 24 25 26 27 28     22 23 24 25 26 27 28
25 26 27 28 29 30 31                              29 30 31

    April                    May                      June

Su Mo Tu We Th Fr Sa     Su Mo Tu We Th Fr Sa     Su Mo Tu We Th Fr Sa
          1  2  3  4                   1  2         1  2  3  4  5  6
 5  6  7  8  9 10 11      3  4  5  6  7  8  9      7  8  9 10 11 12 13
12 13 14 15 16 17 18     10 11 12 13 14 15 16     14 15 16 17 18 19 20
19 20 21 22 23 24 25     17 18 19 20 21 22 23     21 22 23 24 25 26 27
26 27 28 29 30           24 25 26 27 28 29 30     28 29 30
                         31

    July                     August                   September

Su Mo Tu We Th Fr Sa     Su Mo Tu We Th Fr Sa     Su Mo Tu We Th Fr Sa
          1  2  3  4                         1           1  2  3  4  5
 5  6  7  8  9 10 11      2  3  4  5  6  7  8      6  7  8  9 10 11 12
12 13 14 15 16 17 18      9 10 11 12 13 14 15     13 14 15 16 17 18 19
19 20 21 22 23 24 25     16 17 18 19 20 21 22     20 21 22 23 24 25 26
26 27 28 29 30 31        23 24 25 26 27 28 29     27 28 29 30
                         30 31

    October                  November                 December

Su Mo Tu We Th Fr Sa     Su Mo Tu We Th Fr Sa     Su Mo Tu We Th Fr Sa
             1  2  3      1  2  3  4  5  6  7            1  2  3  4  5
 4  5  6  7  8  9 10      8  9 10 11 12 13 14      6  7  8  9 10 11 12
11 12 13 14 15 16 17     15 16 17 18 19 20 21     13 14 15 16 17 18 19
18 19 20 21 22 23 24     22 23 24 25 26 27 28     20 21 22 23 24 25 26
25 26 27 28 29 30 31     29 30                    27 28 29 30 31
```

Deliverables:

1. Source Code: (**.cpp** file) that must start with a comment block similar to the following:

```
/*************************************************************************
** Author          : Suleyman Uludag
** Program          : hw1, q1
** Date Created     : September 15, 2013
** Date Last Modified : September 16, 2013
** Usage            : No command line arguments
**
** Problem:
Accept the following information from the user (keyboard):
- Hw1, hw2 and hw3 (out of 100)
- Midterm (out of 100)
- Final exam (out of 100)
Calculate the total grade out of 100 based on the following grading scale:
Hws          -->    30% (10% each)
Midterm      -->    30%
Final Exam   -->    40%
** Pseudocode:
** 1)
** 2)
*************************************************************************/
```

2. Executable (.exe file under windows). You must explicitly state the platform of your executable (such as Linux, etc.) if it is not Windows. Please name your file by using the question number: **hw1-q1.exe** (for Windows)

3. Screenshot of your app. For screenshot, you can use the following free program on windows:

   `http://www.wisdom-soft.com/downloads/setupscreenhunterfree.exe`

   For Linux/Unix, there are many alternatives. I personally like shutter.
   File naming convention example:

   `hw1_q1.png` (or .jpg or another graphics format)

4. You must zip all the above three files into ONE .zip file and submit your assignment by the deadline on moodle system. Name your file as Lastname-Firstname-hw#.zip. For example, **Uludag-Suleyman-hw1.zip**

   For generating .zip file, you may use the following free software on Windows:

   `http://www.7-zip.org/download.html`

   Linux/Unix has many built-in.