Spring 2024, ISTM, FCU-Purdue 2+2 ECE Program
ISTM2731 , Advanced C Programming, Quiz 2

Total SEVEN FILES for Quiz 2. Use file name quiz2_DXXXXXXX_1.cpp for Question 1 and file names quiz2_DXXXXXXX_2.dev, Node_DXXXXXXX.h, Node_DXXXXXXX.cpp, CStack_DXXXXXXX.h, CStack_DXXXXXXX.cpp, and quiz2_DXXXXXXX_2.cpp for Question 2, where DXXXXXXX is your student ID. When you finish a question, submit the above 7 files to the instructor's computer.

1. (30 points) Start with program skeleton **quiz2_skeleton_1.cpp** and change the file name to **quiz2_DXXXXXXX_1.cpp**. Program skeleton **quiz2_skeleton_1.cpp** contains the specification of **class** Rectangle and two friend functions:
   friend istream & operator>>(istream &, Rectangle &);
   friend ostream & operator<<(ostream &, **const** Rectangle &);.

   Complete the default constructor, member functions and friend functions of **class** Rectangle. Write comments to explain each statement in the implementation of the constructor, member functions, and friend functions. The main function **int** main() is also given the program skeleton. Do **NOT** change specification of **class** Rectangle and the main function.

   Program execution example:

```
Rectangle r1: height 2, width 2, perimeter 8, area 4
          **
          **

----------------------------------------
Rectangle r2: height 5, width 6, perimeter 22, area 30
          ******
          *    *
          *    *
          *    *
          ******

----------------------------------------
Enter height and width of r3: 10 20

Rectangle r3: height 10, width 20, perimeter 60, area 200
          ********************
          *                  *
          *                  *
          *                  *
          *                  *
          *                  *
          *                  *
          *                  *
          *                  *
          ********************
```

(To be continued)

2. (70 points) Change **Node_skeleton.h**, **Node_skeleton.cpp**, **IQueue_skeleton.h**, **IQueue_skeleton.cpp**, and **quize2_skeleton_2.cpp** to **Node_DXXXXXXX.h**, **Node_DXXXXXXX.cpp**, **IQueue_DXXXXXXX.h**, **IQueue_DXXXXXXX.cpp**, and **quiz2_DXXXXXXX_2.cpp**. Create a C++ project **quiz2_DXXXXXXX_2.dev** and add the five .h and .cpp files to the project. Files **Node_DXXXXXXX.h** and **Node_DXXXXXXX.cpp** are the header file and the source file of nodes of a non-circular doubly-linked linear list with integer (**int**) data elements and files **IQueue_DXXXXXXX.h** and **IQueue_DXXXXXXX.cpp** are the header file and the source file of integer queues using non-circular doubly-linked linear list, **quiz2_DXXXXXXX_2.cpp** is the source code of application program performing **YangHui Triangle problem** using an integer queue.

A YangHui Triangle, also known as Pascal's Triangle, is the triangular array of binomial coefficients of polynomial $(a+b)^k$, starting from k=0 to k=n. Example of n=5 is shown below:

```
          1
        1   1
      1   2   1
    1   3   3   1
  1   4   6   4   1
1   5  10  10   5   1
```

Write a C++ project using queue operations to solve YangHui Triangle, also known as Pascal's Triangle, with the algorithm described below. Let Q be a queue.

1. Clear Q;
2. Set k to 0;
3. If k≤n, then
   3.1. Set last and i to 0;
   3.2. If i<k, then
      3.2.1 Set curr = dequeue(Q);
      3.2.2 Print last+curr;
      3.2.3 enqueue(Q, last+curr);
      3.2.4 Set last to curr;
      3.2.5 Increment i by 1;
      3.2.6 Go to Step 3.2;
   3.3 else,
      3.3.1. last = 1;
      3.3.2. print last and a newline;
      3.3.3. enqueue(Q, last);
      3.3.4. Increment k by 1;
      3.3.5. Go to Step 3;
4. else clear Q.

The main program will read an integer n and perform YangHui Triangle algorithm. It will repeat the process until the input integer n is a negative integer. Program execution example:

```
>>>> Enter an integer n (0<=n<=20, stop when n<0): 5
  1
  1   1
  1   2   1
  1   3   3   1
  1   4   6   4   1
  1   5  10  10   5   1
------------------------------
>>>> Enter an integer n (0<=n<=20, stop when n<0): 10
  1
  1   1
  1   2   1
  1   3   3   1
  1   4   6   4   1
  1   5  10  10   5   1
  1   6  15  20  15   6   1
  1   7  21  35  35  21   7   1
  1   8  28  56  70  56  28   8   1
  1   9  36  84 126 126  84  36   9   1
  1  10  45 120 210 252 210 120  45  10   1
------------------------------
>>>> Enter an integer n (0<=n<=20, stop when n<0): -1
```