E2410 Data Structure Hw #2 Part2 (Chapter 3 of textbook)

due date 4/23/2021 by 108061217, 鍾永桓

EECS2040 Data Structure Hw #2 (Chapter 3 Stack/Queue)
due date 4/23/2021

Format: Use a text editor to type your answers to the homework problem. You need to submit your HW in an HTML file or a DOCX, pdf file named as **Hw2-SNo.docx, Hw2-SNo.pdf** or **Hw2-SNo.html**, where SNo is your student number. Send the **Hw2-SNo.doc or Hw2-SNo.html** file in eLearn. Inside the file, you need to put the **header and your student number, name (e.g., EE2410 Data Structure Hw #2 (Chapter 3 of textbook) due date 4/17/2021 by SNo, name)** first, and then the **problem** itself followed by your **answer** to that problem, one by one. The grading will be based on the correctness of your answers to the problems, and the **format**. Fail to comply with the aforementioned format (file name, header, problem, answer, problem, answer,…), will certainly degrade your score. If you have any questions, please feel free to ask me.

Part 2 Coding (50%)

You should submit:
(a) All your source codes (C++ file).
(b) Show the execution trace of your program.

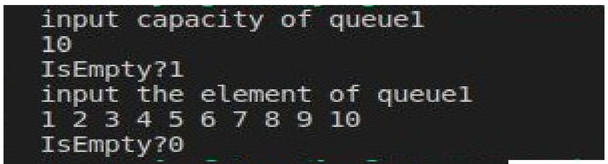
1. (30%) Based on the circular queue and template queue ADT in **ADT 3.2** in textbook (or pptx pp.79), write a C++ program to implement the queue ADT**.** Then add two more functions to
   (a) Return the size and capacity of a queue.
   (b) Merge two queues into a one by alternately taking elements from each queue. Te relative order of queue elements is unchanged. What is the complexity of your function?

   You should **demonstrate all the functions** using at least one example.

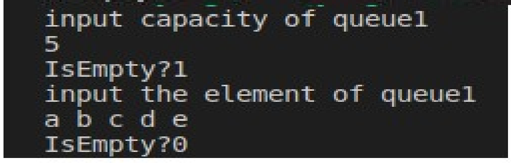
   Ans: To use this program,first we can declare the Queue<type> .And then to add elements to this Queue, we should use Push().Push() can add element to the rear.

To delete the elements, we can use Pop(). We can get the front element by Front() and rear element by Rear() , size by Size() and capacity by Capacity(). If we have to merge to Queue, we can use the Merge().

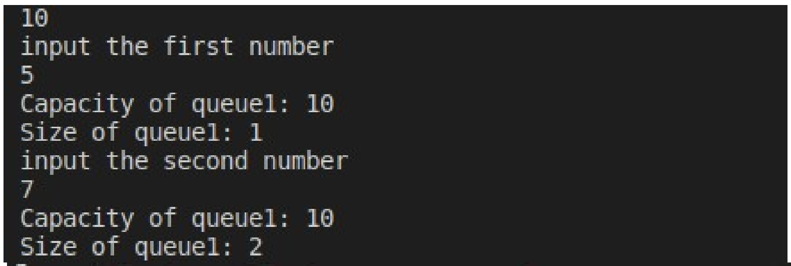
Demo:



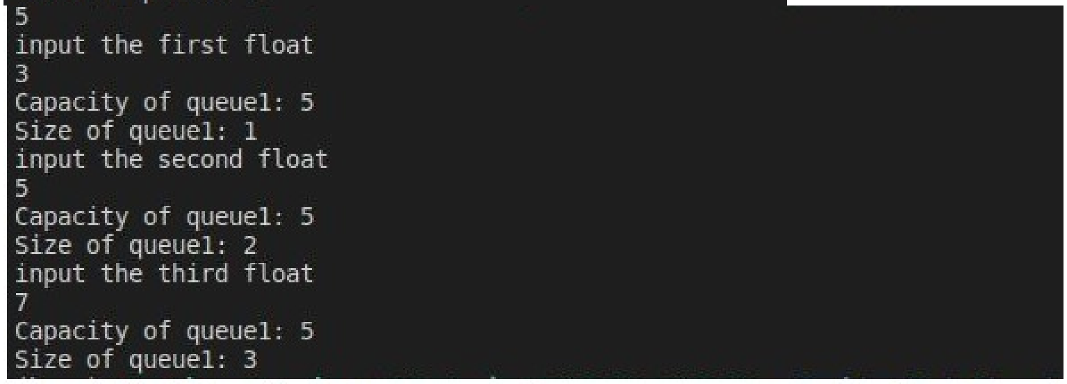
```
input capacity of queue1
10
IsEmpty?1
input the element of queue1
1 2 3 4 5 6 7 8 9 10
IsEmpty?0
```



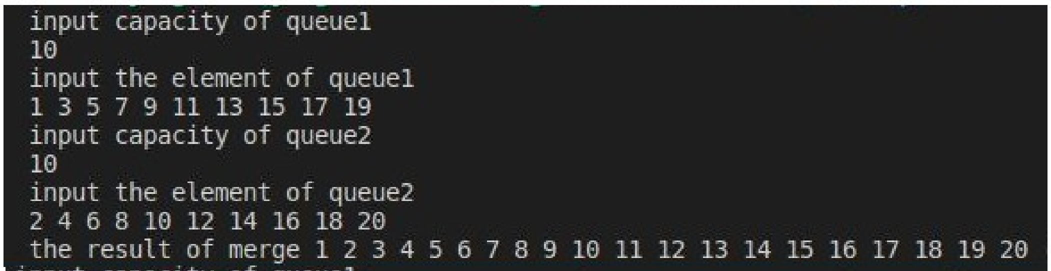
```
input capacity of queue1
5
IsEmpty?1
input the element of queue1
a b c d e
IsEmpty?0
```



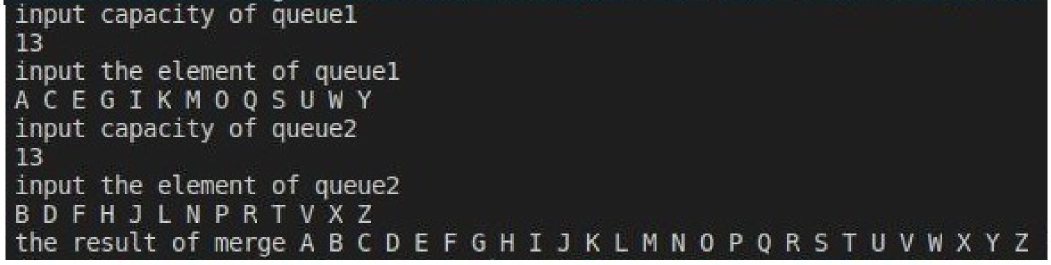
```
10
input the first number
5
Capacity of queue1: 10
Size of queue1: 1
input the second number
7
Capacity of queue1: 10
Size of queue1: 2
```



```
5
input the first float
3
Capacity of queue1: 5
Size of queue1: 1
input the second float
5
Capacity of queue1: 5
Size of queue1: 2
input the third float
7
Capacity of queue1: 5
Size of queue1: 3
```



```
input capacity of queue1
10
input the element of queue1
1 3 5 7 9 11 13 15 17 19
input capacity of queue2
10
input the element of queue2
2 4 6 8 10 12 14 16 18 20
the result of merge 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
```



```
input capacity of queue1
13
input the element of queue1
A C E G I K M O Q S U W Y
input capacity of queue2
13
input the element of queue2
B D F H J L N P R T V X Z
the result of merge A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
```

2. (35%) Referring to **Program 3.13** in textbook (pptx pp.94),

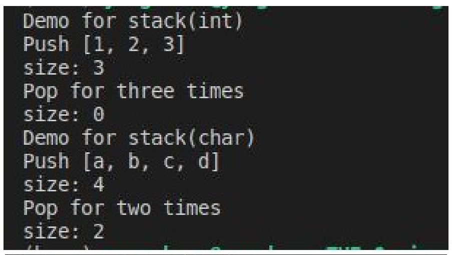
(a) Implement Stack as a publicly derived class of Bag using template. **Demonstrate** your C++ code using at least two element types (e.g., int, float, …). **Show results** of a series of Pushes and Pops and Size functions.

(b) Implement Queue as a publicly derived class of Bag using template. **Demonstrate** your C++ code using at least two element types (e.g., int, float, …). **Show results** of a series of Pushes and Pops and Size functions.

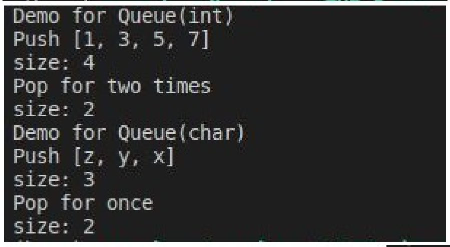
(c) A template double-ended queue (deque) is a linear list in which additions and deletions may be made at either end. Implement the class Deque as a publicly derived templated class of Queue. The class Deque must have public functions (either via inheritance from Queue or by direct implementation in Deque) to add and delete elements from either end of the deque and also to return an element from either end. The complexity of each function (excluding array doubling) should be Θ(1).
Demonstrate your C++ code using at least two element types (e.g., int, float, …). **Show results** of a series of two types of Pushes and Pops and Size functions to illustrate your code is working.

Ans: In the program, we have three kinds of data type (Stack, Queue , and Deque)
To add elements, in Stack ,we use Push() to add element to the top . In Queue we also use Push() , but we add to the rear. In Deque, we can use Push() to add elements to the rear and PushF() to the front. In Stack and Queue , we can only use Pop() to delete element, but we can use Pop() and PopR() in Deque to delete elements in front and rear. In all of the three types, we can use Size() to get size.

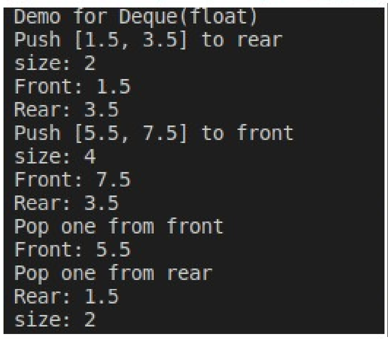
Demo:



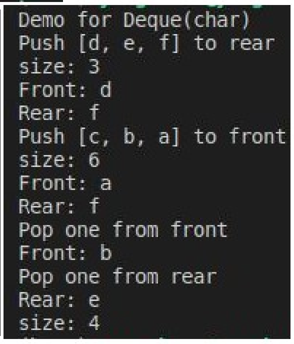
```
Demo for stack(int)
Push [1, 2, 3]
size: 3
Pop for three times
size: 0
Demo for stack(char)
Push [a, b, c, d]
size: 4
Pop for two times
size: 2
```



```
Demo for Queue(int)
Push [1, 3, 5, 7]
size: 4
Pop for two times
size: 2
Demo for Queue(char)
Push [z, y, x]
size: 3
Pop for once
size: 2
```



```
Demo for Deque(float)
Push [1.5, 3.5] to rear
size: 2
Front: 1.5
Rear: 3.5
Push [5.5, 7.5] to front
size: 4
Front: 7.5
Rear: 3.5
Pop one from front
Front: 5.5
Pop one from rear
Rear: 1.5
size: 2
```



```
Demo for Deque(char)
Push [d, e, f] to rear
size: 3
Front: d
Rear: f
Push [c, b, a] to front
size: 6
Front: a
Rear: f
Pop one from front
Front: b
Pop one from rear
Rear: e
size: 4
```

3. (35%) Write a C++ program to implement the maze in textbook using the example codes of **Program 3.15** (pptx pp.106 Algorithm()) and **3.16 (pptx void Path(const int m, const int p)**. You should use a text editor to edit a file containing the maze matrix and then read in the file to establish the maze matrix in your program. The default entrance and exit are located in the upper left corner and lower right corner, respectively as shown in textbook.
   (a) Demonstrate your maze program using the maze shown in **Figure 3.11** in textbook.
   (b) Find a path through the maze shown **Figure 3.14** in textbook.
   (c) Trace out the action of function path (**Program 3.16**) on the maze shown. Compare this to your own attempt in (b).

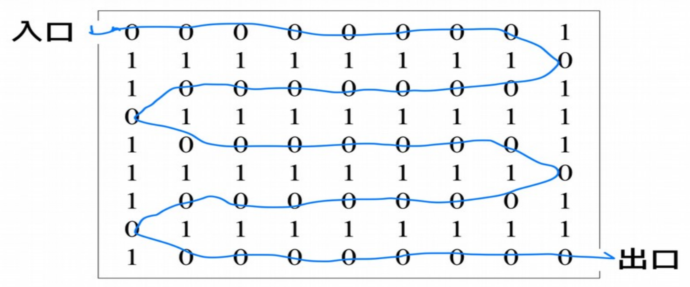
```
入口  0 1 0 0 0 1 1 0 0 0 1 1 1 1 1
      1 0 0 0 1 1 0 1 1 1 0 0 1 1 1
      0 1 1 0 0 0 0 1 1 1 1 0 0 1 1
      1 1 0 1 1 1 1 0 1 1 0 1 1 0 0
      1 1 0 1 0 0 1 0 1 1 1 1 1 1 1
      0 0 1 1 0 1 1 1 0 1 0 0 1 0 1
      0 0 1 1 0 1 1 1 0 1 0 0 1 0 1
      0 1 1 1 1 0 0 1 1 1 1 1 1 1 1
      0 0 1 1 0 1 1 0 1 1 1 1 1 0 1
      1 1 0 0 0 1 1 0 1 1 0 0 0 0 0
      0 0 1 1 1 1 1 0 0 0 1 1 1 1 0  出口
      0 1 0 0 1 1 1 1 1 0 1 1 1 1 0
```

Figure 3.11：一個迷宮的例子（你能找出一條路徑嗎？）

```
入口  0 0 0 0 0 0 0 0 1
      1 1 1 1 1 1 1 1 0
      1 0 0 0 0 0 0 0 1
      0 1 1 1 1 1 1 1 1
      1 0 0 0 0 0 0 0 1
      1 1 1 1 1 1 1 1 0
      1 0 0 0 0 0 0 0 1
      0 1 1 1 1 1 1 1 1
      1 0 0 0 0 0 0 0 0  出口
```

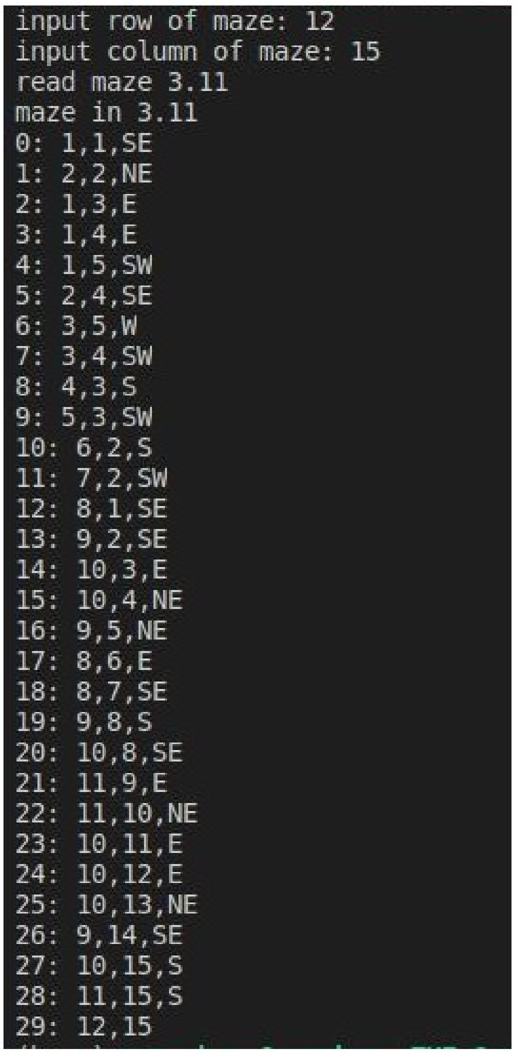
Figure 3.14：唯一路徑的迷宮圖

Ans: In the program, we input the row and column of the maze first. And then we can read the matrix of maze from other file. We use Path(row of row, col of destination) to output the path to the destination.

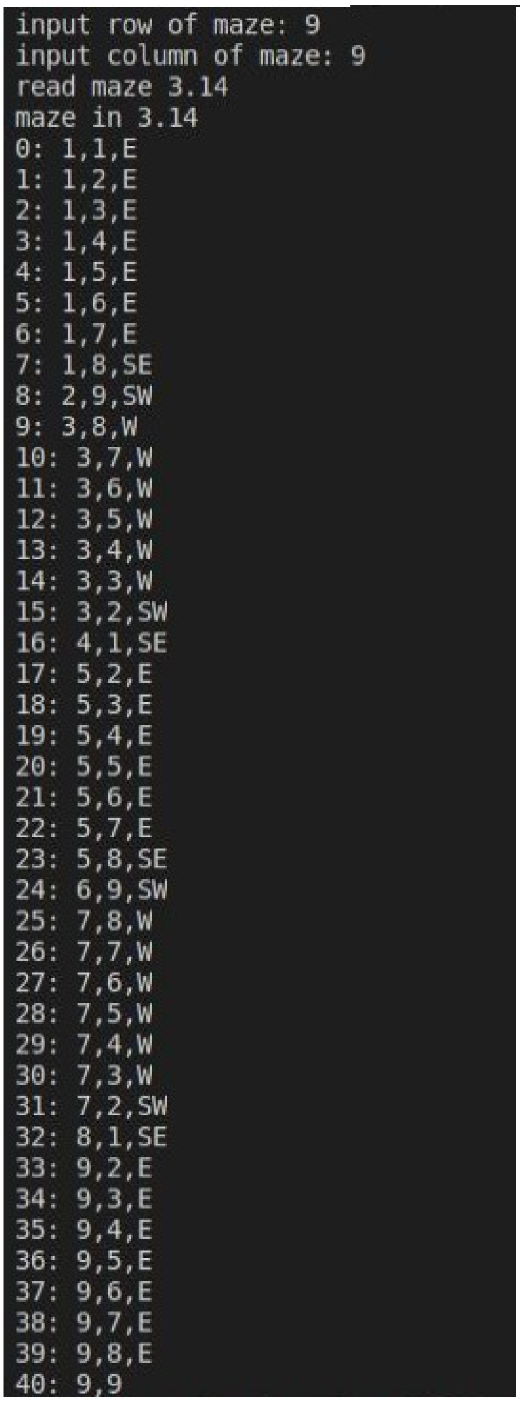


```
0 0 0 0 0 0 0 0 1
1 1 1 1 1 1 1 1 0
1 0 0 0 0 0 0 0 1
0 1 1 1 1 1 1 1 1
1 0 0 0 0 0 0 0 1
1 1 1 1 1 1 1 1 0
1 0 0 0 0 0 0 0 1
0 1 1 1 1 1 1 1 1
1 0 0 0 0 0 0 0 0
```

Demo:



```
input row of maze: 12
input column of maze: 15
read maze 3.11
maze in 3.11
0: 1,1,SE
1: 2,2,NE
2: 1,3,E
3: 1,4,E
4: 1,5,SW
5: 2,4,SE
6: 3,5,W
7: 3,4,SW
8: 4,3,S
9: 5,3,SW
10: 6,2,S
11: 7,2,SW
12: 8,1,SE
13: 9,2,SE
14: 10,3,E
15: 10,4,NE
16: 9,5,NE
17: 8,6,E
18: 8,7,SE
19: 9,8,S
20: 10,8,SE
21: 11,9,E
22: 11,10,NE
23: 10,11,E
24: 10,12,E
25: 10,13,NE
26: 9,14,SE
27: 10,15,S
28: 11,15,S
29: 12,15
```



```
input row of maze: 9
input column of maze: 9
read maze 3.14
maze in 3.14
0: 1,1,E
1: 1,2,E
2: 1,3,E
3: 1,4,E
4: 1,5,E
5: 1,6,E
6: 1,7,E
7: 1,8,SE
8: 2,9,SW
9: 3,8,W
10: 3,7,W
11: 3,6,W
12: 3,5,W
13: 3,4,W
14: 3,3,W
15: 3,2,SW
16: 4,1,SE
17: 5,2,E
18: 5,3,E
19: 5,4,E
20: 5,5,E
21: 5,6,E
22: 5,7,E
23: 5,8,SE
24: 6,9,SW
25: 7,8,W
26: 7,7,W
27: 7,6,W
28: 7,5,W
29: 7,4,W
30: 7,3,W
31: 7,2,SW
32: 8,1,SE
33: 9,2,E
34: 9,3,E
35: 9,4,E
36: 9,5,E
37: 9,6,E
38: 9,7,E
39: 9,8,E
40: 9,9
```