



# University of Central Punjab

## ADP

### Data Structures and Algorithms

Fall 2021

Section – ADP32

#### Assignment # 2

**Question #1:** Consider the following statements:

```
stackType<int> stack;
```

```
int x;
```

Suppose that the input is:

14 45 34 23 10 5 -999

Show what is output by the following segment of code:

```
stack.push(5);
```

```
cin >> x;
```

```
while (x != -999)
```

```
{ if (x % 2 == 0)
```

```
{ if (!stack.isFullStack())
```

```
stack.push(x); }
```

```
else
```

```
cout << "x = " << x << endl;
```

```
cin >> x; }
```

```
cout << "Stack Elements: ";
```

```
while (!stack.isEmptyStack())
```

```
{ cout << " " << stack.top();
```

```
stack.pop(); }
```

```
cout << endl;
```

**Question #2:** Evaluate the following postfix expressions:

a.  $8\ 2\ +\ 3\ *\ 16\ 4\ /\ -\ =$

b.  $12\ 25\ 5\ 1\ /\ *\ 8\ 7\ +\ -\ =$

c.  $70\ 14\ 4\ 5\ 15\ 3\ /\ * - - /\ 6\ + =$

d.  $3\ 5\ 6\ *\ +\ 13 - 18\ 2\ /\ + =$

**Question #3:** Convert the following infix expressions to postfix notations:

a.  $(A + B) * (C + D) - E$

b.  $A - (B + C) * D + E / F$

c.  $((A + B) / (C - D) + E) * F - G$

d.  $A + B * (C + D) - E / F * G + H$

**Question #4:** Write the equivalent infix expression for the following postfix expressions:

a.  $A\ B\ *\ C\ +$

b.  $A\ B\ +\ C\ D\ -\ *$

c.  $A\ B\ -\ C\ -\ D\ *$

**Question #5:** What is the output of the following program?

```
#include <iostream>
#include <string>
#include "myStack.h"
using namespace std;
void mystery(stackType<int>& s, stackType<int>& t);
int main()
{ int list[] = {5, 10, 15, 20, 25};
  stackType<int> s1;
  stackType<int> s2;
  for (int i = 0; i < 5; i++)
    s1.push(list[i]);
  mystery(s1, s2);
  while (!s2.isEmptyStack())
  { cout << s2.top() << " ";
    s2.pop();}
  cout << endl; }
void mystery(stackType<int>& s, stackType<int>& t)
{ while (!s.isEmptyStack())
  { t.push(2 * s.top());
    s.pop(); }}
```

**Question #6:** Consider the following statements:

```
queueType<int> queue;
```

```
int x, y;
```

Show what is output by the following segment of code:

```
x = 4;
```

```
y = 5;
```

```
queue.addQueue(x);
```

```
queue.addQueue(y);
```

```
x = queue.front();
```

```
queue.deleteQueue();
```

```
queue.addQueue(x + 5);
```

```
queue.addQueue(16);
```

```
queue.addQueue(x);
```

```
queue.addQueue(y - 3);
```

```
cout << "Queue Elements: ";
```

```
while (!queue.isEmptyQueue())
```

```
{ cout << queue.front() << " ";
```

```
queue.deleteQueue(); }
```

```
cout << endl;
```

**Question #7:** What does the following function do?

```
void mystery(queueType<int>& q)
```

```
{ stackType<int> s;
```

```
while (!q.isEmptyQueue())
```

```
{ s.push(q.front());
```

```
q.deleteQueue();}
```

```
while (!s.isEmptyStack())
```

```
{ q.addQueue(2 * s.top());
```

```
s.pop();
```

```
}
```

```
}
```

**Question #8:** Suppose that queue is a queueType object and the size of the array implementing queue is 100. Also, suppose that the value of queueFront is 50 and the value of queueRear is 99.

- a. What are the values of queueFront and queueRear after adding an element to queue?
- b. What are the values of queueFront and queueRear after removing an element from queue?

**Question #9:** Suppose that queue is a queueType object and the size of the array implementing queue is 100. Also, suppose that the value of queueFront is 99 and the value of queueRear is 25.

What are the values of queueFront and queueRear after adding an element to queue?

- b. What are the values of queueFront and queueRear after removing an element from queue?

**Question# 10:**

Consider the following statements:

```
stackType<int> stack;  
queueType<int> queue;  
int x;
```

Suppose the input is:

15 28 14 22 64 35 19 32 7 11 13 30 -999

Show what is written by the following segment of code:

```
stack.push(0);  
queue.addQueue(0);  
cin >> x;  
while (x != -999)  
{ switch (x % 4)  
{ case 0:  
stack.push(x);  
break;  
case 1:  
if (!stack.isEmptyStack())
```

```

{ cout << "Stack Element = " << stack.top()
<< endl;
stack.pop();}
else
cout << "Sorry, the stack is empty." << endl;
break;
case 2:
queue.addQueue(x);
break;
case 3:
if (!queue.isEmptyQueue())
{ cout << "Queue Element = " << queue.front()
<< endl;
queue.deleteQueue();
}else
cout << "Sorry, the queue is empty." << endl;
break;
} //end switch
cin >> x;
} //end while
cout << "Stack Elements: ";
while (!stack.isEmptyStack())
{cout << stack.top() << " ";
stack.pop();
} cout << endl;
cout << "Queue Elements: ";
while (!queue.isEmptyQueue())
{cout << queue.front() << " ";
queue.deleteQueue();
}cout << endl;

```