Chapter 17 – Exercises – solutions to odd number questions

1. a.true; b. false; c. false; d. true; e. false; f. true; g. false; h. false; i. true; j. true;

   k. true; l. false; m. false; n. false;

3. a. 8 b. 7 c. `dec = stack.top();` d. `stack.pop();`

5. ```
   13
   32 32 13 16 28
   temp = 16
   ```

7. ```
   secretNum = 226
   ```

9. a. `16`

   b. `-4`

   c. `39`

   d. `12`

   e. `15`

11. a. `x * y + z - t`

    b. `x * (y + z) - w / u`

    c. `(x - y) * (z / u) - (t + s)`

    d. `x * (y - (z + w))`

13. `1 16 27 16 5`

15. If the stack is nonempty, the statement `stack.top();` returns the top element of the stack and the statement `stack.pop();` removes the top element of the stack.

17. ```
    template <class elemType>
    elemType second(stackType<elemType> stack)
    {
        elemType temp1, temp2;

        if (stack.isEmptyStack())
        {
            cout << "Stack is empty." << endl;
            exit(0); //terminate the program
        }

        temp1 = stack.top();
        stack.pop();

        if (stack.isEmptyStack())
        {
            cout << "Stack has only one element." << endl;
            exit(0); //terminate the program
        }

        temp2 = stack.top();
        stack.push(temp1);

        return temp2;
    }
    ```
    ```

19. a. 4

b. 21

c. `!queue.isEmptyQueue()`

d. `queue.addQueue("programming")`

After the insertion operation the index of the last element is `5`

21.
```
cin >> num;

while (cin)
{
    switch (num % 2)
    {
    case 0:
        stack.push(num);
        break;
    case 1: case -1:
        if (num % 3 == 0)
            queue.addQueue(num);
        else
        {
            if (!stack.isEmptyStack())
                stack.pop();
            stack.push(num * num);
        }
    } //end switch

    cin >> num;
} //end while
```

After processing these numbers, `stack` and `queue` are:

```
stack: 14 289 10 121 28
queue: 15 -9 21 -3 33
```

23. a. 26
b. `queueFront = 35; queueRear = 61.`
c. `queueFront = 36; queueRear = 60.`

25. a. 31
b. `queueFront = 25; queueRear = 56.`
c. `queueFront = 26; queueRear = 55.`

27. 51

29. `5 -4 5 -7 1 2 1 4 1 -2 2 -7 7 -6`

31.
```
template <class Type>
void reverseStack(stackType<Type> &s)
{
    linkedQueueType<Type> q;
    Type elem;
```

```
    while (!s.isEmptyStack())
    {
        elem = s.top();
        s.pop();
        q.addQueue(elem);
    }

    while (!q.isEmptyQueue())
    {
        elem = q.front();
        q.deleteQueue();
        s.push(elem);
    }
}
```

33.

```
template <class Type>
int queueType<Type>::queueCount()
{
    return count;
}
```

35.