Some simple exercises on stacks and queues

1. Present the output in its content and form when the following piece of code is executed:

```cpp
IntStack st;
int x, y;
st.initialize();
x = 4;
y = 6;
st.push(7);
st.push(x);
st.push(x+5);
y = st.top();
st.pop();
st.push(x+y);
st.push(y-2);
st.push(3);
x = st.top();
st.pop();
cout << "x = " << x << endl;
cout << "y = " << y << endl;
while(!st.isEmpty())
{
    y = st.top();
st.pop();
cout << y << endl;
}
```
2. Present the output in its content and form when the piece of code below is executed on the following input: 14 45 34 23 10 5 -999

```cpp
IntStack st;
int x, y;
st.initialize();
st.push(5);
cin >> x;
while(x != -999)
{
    if(x%2 == 0)
        st.push(x);
    else
        cout << "x = " << x << endl;
    cin >> x;
}
cout << "Stack elements: ";
while(!st.isEmpty())
{
    y = st.top();
    st.pop();
    cout << "  " << y;
}
cout << endl;
```

3. Present the output in its content and form when the following piece of code is executed:

```cpp
IntQ q;
int x, y;
q.initialize();
x = 4; y = 5;
q.enqueue(x);
q.enqueue(y);
x = q.getFront();
q.dequeue();
q.enqueue(x+5);
q.enqueue(16);
q.enqueue(x);
q.enqueue(y-3);
cout << "Queue elements: ";
while(!q.isEmpty())
{
    y = q.getFront();
    q.dequeue();
    cout << "  " << y;
}
```
4. Present the output in its content and form when the piece of code below is executed on the following input: 15 28 14 22 19 21 -999

```cpp
IntQ q; // A queue of integers
int x, y;
cin >> x;
while (x != -999)
{
    switch(x % 4)
    {
    case 0: case 1: case 2: q.enqueue(x);
        break;
    case 3: if(!q.isEmpty())
        {
            y = q.getFront(); q.dequeue();
            cout << "Queue element = " << y << endl;
        }
        else
            cout << "Sorry, queue is empty " << endl;
        break;
    } // end of switch
    cin >> x;
} // end while
cout << endl << "Queue elements: ";
while (!q.isEmpty())
{
    x = q.getFront(); q.dequeue();
    cout << x << " ";
} //end of while
cout << endl;
```