```cpp
#include <iostream>
#include <stdlib.h> //srand, rand
#include <time.h>//clock_t, clock, CLOCKS_PER_SEC
using namespace std;

// implementing the Queue ADT using dynamic array


class Queue{

    private:
        int *array;
        int maxSize; // useful to decide if resizing (doubling the array size) is needed
        int endOfQueue; // same as endOfArray

    public:
        Queue(int size){
            array = new int[size];
            maxSize = size;
            endOfQueue = -1;
        }

        bool isEmpty(){

            if (endOfQueue == -1)
                return true;

            return false;
        }

        void resize(int s){

            int *tempArray = array;

            array = new int[s];

            for (int index = 0; index < min(s, endOfQueue+1); index++){
                array[index] = tempArray[index];
            }

            maxSize = s;

        }


        void enqueue(int data){ // same as insert 'at the end'

            if (endOfQueue == maxSize-1)
                resize(2*maxSize);

            array[++endOfQueue] = data;

        }


        int peek(){

            if (endOfQueue >= 0)
                return array[0];
            else
                return -1000000;// an invalid value indicating
                                // queue is empty

        }
```

```
65
66
67          int dequeue(){
68
69              if (endOfQueue >= 0){
70                  int returnVal = array[0];
71
72                  for (int index = 0; index < endOfQueue; index++)
73                      array[index] = array[index+1];
74
75                  endOfQueue--;
76                  // the endOfQueue is decreased by one
77
78                  return returnVal;
79              }
80              else
81                  return -1000000; // an invalid value indicating
82                                   // queue is empty
83          }
84
85
86
87
88
89  };
90
91  int main(){
92
93      Queue queue(1);
94
95      int queueSize;
96
97      cout << "Enter the number of elements you want to enqueue: ";
98      cin >> queueSize;
99
100     srand(time(NULL));
101
102     int maxValue;
103
104     cout << "Enter the maximum value for an element: ";
105     cin >> maxValue;
106
107     cout << "Elements enqueued: ";
108     for (int i = 0; i < queueSize; i++){
109
110         int value = rand() % maxValue;
111         queue.enqueue(value);
112         cout << value << " ";
113     }
114
115     cout << endl;
116
117     cout << "Elements dequeued: ";
118     while (!queue.isEmpty()){
119
120         cout << queue.dequeue() << " ";
121     }
122
123     cout << endl;
124
125 return 0;
126 }
```

```
Enter the number of elements you want to enqueue: 10
Enter the maximum value for an element: 50
Elements enqueued: 13 17 25 40 4 45 21 0 3 4
Elements dequeued: 13 17 25 40 4 45 21 0 3 4
```