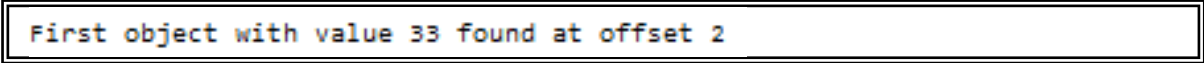


Notice: Return your C++ source codes into Oma before deadline. Only so you can get credits from this exercise. You can't return these source codes after deadline.

sources: <http://users.metropolia.fi/~pasitr/2017-2018/TI00AA50-3011/tt/06/STL.pdf>

- <http://www.cplusplus.com/reference/stl/>
- <http://www.cplusplus.com/reference/iterator/>
- <http://www.cplusplus.com/reference/algorithm/>

1. **Find.** Implement a program **1.cpp** which contains an array `int arr[] = { 11, 22, 33, 44, 55, 66, 77, 88 };`. Program must clarify the offset of number 33. You have to use right function (that is you job). Sample output is in figure 1. [Laf02, p. 735]



```
First object with value 33 found at offset 2
```

Figure 1. Sample print in Dev C++ -program

2. **Count.** Implement a program **2.cpp** which contains an array `int arr[] = { 33, 22, 33, 44, 33, 55, 66, 77 };`. Program must clarify how many times 33 is found in array. You have to use right function (that is you job). Sample output is in figure 2. [Laf02, p. 736]



```
There are 3 33's in arr.
```

Figure 2. Sample print in Dev C++ -program

3. **Sort.** Implement a program **3.cpp** which contains an array `int arr[] = {45, 2, 22, -17, 0, -30, 25, 55};`. Program must sort an array and after that print an array. You have to use right function (that is you job). Sample output is in figure 3. [Laf02, p. 737]



```
-30 -17 0 2 22 25 45 55
```

Figure 3. Sample print in Dev C++ -program

4. **Search.** Implement a program **4.cpp** which contains arrays `int source[] = { 11, 44, 33, 11, 22, 33, 11, 22, 44 };` and `int pattern[] = { 11, 22, 33 };`. Program must clarify the offset of sequence 11, 22, 33. You have to use right function (that is you job). Sample output is in figure 4. [Laf02, p. 737]



```
Match at 3
```

Figure 4. Sample print in Dev C++ -program

5. **Merge.** Implement a program **5.cpp** which contains three arrays **int src1[] = { 2, 3, 4, 6, 8 };**, **int src2[] = { 1, 3, 5 };** and **int dest[8];**. Program merges the elements from two source containers into a destination. You have to use right function (that is your job). Sample output is in figure 5. [Laf02, p. 738]

```
1 2 3 3 4 5 6 8
```

Figure 5. Sample print in Dev C++ -program

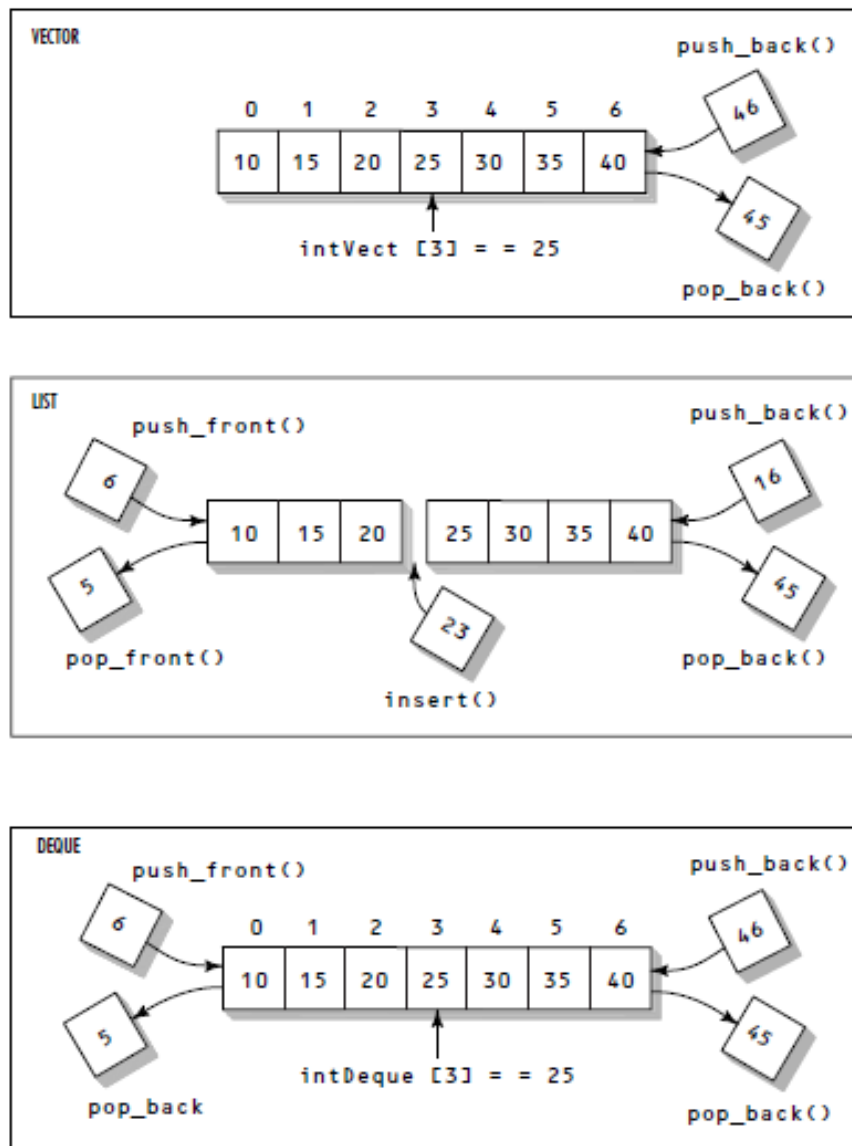


Figure 6. Ideas of Vector, List and Deque you must use in next tasks

< continued >

6. **Vector, push_back** and **size**. You can see the idea of Vector in figure 6. Implement a program **6.cpp** where you define vector **v**. After that put values 10, 11, 12 and 13 at the end of array. Replace first value and third value of array so that first value is 20 and third value is 23. Finally display vector contents (figure 7). You have to use right functions (that is your job). [Laf02, p. 744]

```
20 11 12 23
```

Figure 7. Sample print in Dev C++ -program

7. **Vector, insert** and **erase**. Implement a program **7.cpp** where you define array **int arr[] = { 100, 110, 120, 130 };**. After that put value 115 at the middle of array. Further erase element 2. You have to use right functions (that is your job). Finally display vector contents (figure 8). [Laf02, p. 746]

```
Before insertion: 100 110 120 130
After insertion:  100 110 115 120 130
After erasure:    100 110 120 130
```

Figure 8. Sample print in Dev C++ -program

8. **Lists**. An STL list container is a doubly linked list, in which each element contains a pointer not only to the next element but also to the preceding one (figure 6). Implement a program **8.cpp** where you define a list with clause **list<int> ilist;** and push back values 30 and 40. After that push front values 20 and 10. Finally print all values from front. Sample print is in figure 9. [Laf02, p. 747]

```
10 20 30 40
```

Figure 9. Sample print in Dev C++ -program

9. **Deque**. You can see the idea of Vector in figure 6. A deque is like a vector in some ways and like a linked list in others. Like a vector, it supports random access using the **[]** operator. However, like a list, a deque can be accessed at the front as well as the back. It's a sort of double-ended vector, supporting **push_front()**, **pop_front()**, and **front()**. Implement a program **9.cpp** where you define a list with clause **deque<int> deq;** and push back values 30, 40 and 50. After that push front values 20 and 10. Then you have to change middle item (value 33). Finally display all items. Sample print is in figure 10. [Laf02, p. 750]

```
10 20 33 40 50
```

Figure 10. Sample print in Dev C++ -program