

Homework 5

Assigned in Laboratory 8

Due at Start of Laboratory 9

This is a pledged assignment; you may receive help only from the class teaching assistants, ITC consultants, and the class professors. You may work in groups of up to two for this assignment. Members of a group must come from the same lab section. No code can be shared between groups.

Objective

The objective of this assignment is to teach you about user-defined functions and sequences.

Assignment files

A self-extracting archive `hw05.exe` has been created for this assignment. The archive will produce files `funcfun.i.de`, `hw05.cpp`, `main.cpp`, `sequence.h`, and `seq.lib`. Archive `hw05.exe` can be acquired by accessing the course website <http://www.cs.virginia.edu/cs101>. If you are copying this file using a web browser, it is recommended that you first copy the file to a hard drive and then, if necessary, copy the file to a floppy. The file when completed will produce a file `funcfun.exe` that can be run.

Problem Description

A sequence is an ordered-list of values. Thus we can talk about the i^{th} element of a sequence, where i can vary between 1 and the number of elements in the sequence. Also, a value may occur more than once within a sequence. For example, `[10 21 5 17 21 3 21]` is a sequence of seven elements, where the second, fifth and seventh elements all have the same value of 21. For our purposes, the values within a sequence are constrained to be positive integers.

A class `Sequence` has been defined for you. The header file for this class is `sequence.h`. The implementation for the `Sequence` is provided for you in library form. The library is named `seq.lib`.

The class `Sequence` has a single constructor that acts as the default constructor. This constructor creates an empty sequence. Thus, the following example defines a `Sequence` object `S`, where `S` starts off with no elements in its sequence.

```
Sequence S;
```

In addition, the class `Sequence` has the following member functions.

- `void set(int i, int v)`
Sets the i^{th} element of a sequence to v . For example, if `T` represents the sequence `[6 21]`, then the invocation `T.set(2, 9)` would update `T`'s representation to be `[6 9]`. Note, if there is no i^{th} element, the program terminates.
- `int peek(int i)`
Returns the value of the i^{th} element of the sequence. For example, if `T` represents the sequence `[6 21]`, then the invocation `T.peek(1)` returns the value 6. Note: if there is no i^{th} element, the program terminates.
- `int size()`
Returns the number of elements in the sequence. For example, if `T` represents the sequence `[6 21]`, then the invocation `T.size()` returns 2.
- `void push_back(int v):`
Adds the value v to the end of the sequence. For example, if `T` represents the sequence `[6 21]`, then the invocation `T.push_back(54)` would update `T`'s representation to be `[6 21 54]`.

- `void erase(int i)`
Removes the i^{th} member of the sequence and move all members to the right of the i^{th} member to the left one position, reducing the size of the sequence by one. For example, if `T` represents the sequence `[6 21 54]`, then the invocation `T.erase(1)` would update `T`'s representation to be `[21 54]`. Note: if there is no i^{th} element, the program terminates.

The insertion operator has been overloaded for Sequence objects. Thus, if `T` is an object representing the sequence `[6 21 54]`, the statement

```
cout << T << endl;
```

Would produce as its output.

```
[ 6 21 54 ]
```

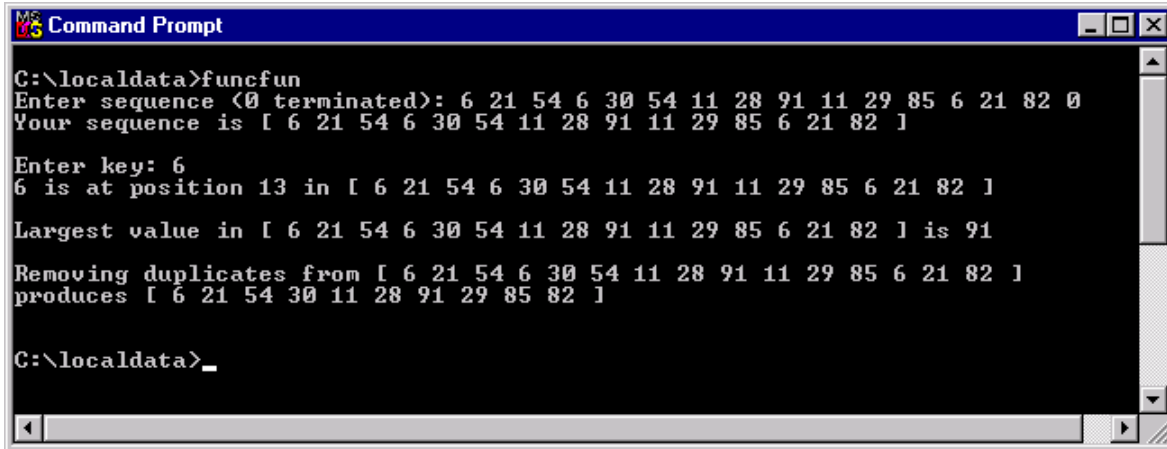
Your assignment is to write several functions for the file `hw05.cpp` that operate on sequences.

- `void ExtractTo(Sequence &S)`
Prompts the user for values to append to the sequence `S`. Function `ExtractTo()` should stop appending values once the user enters a 0 (the 0 should not be part of the sequence).
- `int Find(const Sequence &S, int v)`
Returns the position of the last occurrence of the value `v` in the sequence `S`. If `S` does not contain the value `v`, the function returns 0.
- `int Max(const Sequence &S)`
Returns the value of the largest element in the sequence `S`. If `S` is an empty sequence, the function instead displays an error message and terminates the program immediately. The `std::ib` function `exit()` should prove useful here.
- `void RemoveDuplicates(Sequence &S)`
Duplicates of earlier occurring values are removed from the list. The leftmost (smallest valued index) occurrence of a repeated element should be the one left in the sequence. Thus, in the sequence `[11 28 11 29 6 21 6 9]`, the third element should be removed as its value is a duplicate of the value of the first element. Similarly, the seventh element should be removed as its value is a duplicate of the value of the fifth element. Thus, the resulting sequence should be `[11 28 29 6 21 9]`.

Hint: the following process is suggested for `RemoveDuplicates()`

```
for i in the interval 1 to S.size() do
    determine the value v which is the ith element of S
    determine the location j which is the last location in S that
    contains the value v
    while j and i are different repeatedly erase the
    occurrence of v from S at position j and then update j so that
    it is again the last location in S that contains the value v
end for
```

Your functions should be placed in the file `hw05.cpp`. This file is part of the project `funcfun.ide`. You will also find as part of the project a file `main.cpp`. This file will contain a function `main()` that allows us to test your functions. For example, the following input/output behavior



```

C:\localdata>funcfun
Enter sequence (0 terminated): 6 21 54 6 30 54 11 28 91 11 29 85 6 21 82 0
Your sequence is [ 6 21 54 6 30 54 11 28 91 11 29 85 6 21 82 ]

Enter key: 6
6 is at position 13 in [ 6 21 54 6 30 54 11 28 91 11 29 85 6 21 82 ]

Largest value in [ 6 21 54 6 30 54 11 28 91 11 29 85 6 21 82 ] is 91

Removing duplicates from [ 6 21 54 6 30 54 11 28 91 11 29 85 6 21 82 ]
produces [ 6 21 54 30 11 28 91 29 85 82 ]

C:\localdata>_

```

could be produced by the following function `main()`. During our testing we may use a different function `main()`!

```

int main() {
    Sequence S1;
    ExtractTo(S1);
    cout << "Your sequence is " << S1 << endl << endl;
    cout << "Enter key: " << flush;
    int key;
    cin >> key;
    if (Find(S1, key) != 0) {
        cout << key << " is at position " << Find(S1, key) << " in " << S1
            << endl << endl;
    }
    else {
        cout << S1 << " does not contain " << key << endl << endl;
    }
    cout << "Largest value in " << S1 << " is " << Max(S1) << endl
        << endl;
    cout << "Removing duplicates from " << S1 << endl;
    RemoveDuplicates(S1);
    cout << "produces " << S1 << endl << endl;
    return 0;
}

```

Notes

- When working on a project, the project file must be opened *before* the source file(s) are opened. Therefore, when working on this assignment, open `funcfun.ide` *before* opening `hw05.cpp`. To open an `ide` file, select the item `Open Project` under Borland's `Project` menu.
- To open `hw05.cpp` for editing, double click on its filename in the project window (this window is typically displayed in the bottom of the IDE). *Remember to save your program frequently.*

- You should run your program from a command window. The name of the program will be `funcfun.exe`. It will be placed by the project in `c:\local data`.
- The project file for this assignment was developed for use in one of the on-grounds ITC computer laboratories such as the Catlin Laboratory (The Stacks). In particular, as required by the Borland compiler, the project file specifies the locations of local and standard libraries. Our specification is their location on the ITC servers. If you decide to work on this assignment elsewhere, please refer to the handout available in the documents section of the class web site.
- Only the file `hw05.cpp` should be submitted electronically before the laboratory in which it is due. You should also turn in a hardcopy of this file at the beginning of the laboratory in which it is due.
- The program should follow course programming style guidelines. The guideline is available on the CS 101 website. In particular named constants should be used to represent the number of data values and the data value interval. Be sure that you consistently use the named constants rather than literal values in your program.
- The grading criteria for this assignment will be made available on the CS 101 website.