Algorithms

Lab 02 (8%)

Topics: class VecInt, destructor,
deep and shallow copies, copy constructor and assignment operator.
(Reading: Stroustrup “Programming Principles and Practicies”. Chapters 17, 18)

## Problem 01 (0.5%)
(advantages of C++ classes)

Solve last problem of previous lab using standard class vector. Explain potential problems of using new type VecInt in C-programs and advantages C++ classes’ mechanism gives to define new types.

## Problem 02 (2%)
(start of developing class VecInt, constructors, destructors, index operator, method pushBack)

Create initial version of class VecInt to solve Problem 01.
Your version has to have:

- default constructor: creates empty vector;
- destructor: releases dynamic memory which vector owns;
- method int* begin(): returns ponter to the beginning of vector;
- method int* end(): returns pointer to the end of vector;
- method pushBack: adds element to the end of the vector;

## Problem 03 (0.5%)
(deep copy vs shallow copy)

Write a simple C++ program using standard class vector and standard algorithm reverse. Program has to read arbitrary sequence of integer numbers and check whether this sequence is a palindrome or not.

You have to

- read all integer numbers from standard vector v;
- create a vector w using v as initializer;
- reverse w using std::reverse;
- compare v and w and if they are equal print “Palindrome”, otherwise “not palindrome”;

## Problem 04 (2.5%)
(class VecInt, deep vs shallow copy, iterators)

Try to solve Problem 03 using class VecInt from Problem 02. Explain runtime error you have.

- add copy constructor;
- add assignment operator;
- add method size() and subscript operators;
- add operator==
- add operator!=

Solve Problem 03 using improved version of class VecInt and std::reverse.

Problem 05(2.5%)

(class VecInt, methods insert and erase)

Add methods (emulating std::vector::insert and std::vector::erase methods)

- int* insert(int* pos, int e)
- int* erase(int* pos)

Using improved version of your class to solve following program:

- read arbitrary amount of integer numbers into VecInt
- insert before each even number value 0;
- print the contexts of VecInt;
- erase all even numbers from VecInt;
- print the contexts of VecInt;