Dirk Praetorius,
Thomas Führer

Sommersemester 2014
20.05.2014

Übungen zur Vorlesung
Einführung in das Programmieren für TM

Serie 10

Aufgabe 10.1. Write a function `unique` which sorts a vector in ascending order, eliminates entries that appear more than once, and returns the shortened vector. Use the C++-standard container `vector`. What are the differences with your first implementation?

Aufgabe 10.2. Implement the simple *Tic Tac Toe* game. The rules can be found at

> http://en.wikipedia.org/wiki/Tic-tac-toe

Your program should fulfill the following criteria:

1. Two players who could play against each other.
2. An automatic check if one of the player has won.
3. Print out the playfield after each step.
4. If none of the player can win, print out `'tied game'`

Use the container `vector` for the playfield, e.g., `vector<vector<char> >`. Each of the player should use its own character symbol, e.g., `'x'` for player 1 and `'o'` for player 2. The playfield should be an $n \times n$-array with $n \geq 3$!

Aufgabe 10.3. The encryption algorithm *ROT13* "is a simple letter substitution cipher that replaces a letter with the letter 13 letters after it in the alphabet" (see Wikipedia). This means that the letters A,B,C ...are replaced by N,O,P..., and the letters N,O,P by A,B,C. Write a function `rot13` which implements this algorithm. Your function just needs to handle capital letters. Then, generalize the idea of the algorithm, and write a function `rot_n` that rotates the letters by n places, not only by 13. Moreover, write a function `unrot_n`, which deciphers an encrypted text. *Hint:* A character is of typ `char` which is internally handled by a number, the character `'A'` for instance is given by the number 65 (in ASCII code). If you substract `'A'` from one of the letters `'A'`,...,`'Z'` the resulting number is between 0 and 25.

Aufgabe 10.4. Write a class `Matrix` for the storage of $m \times n$ matrices $A \in \mathbb{R}^{m\times n}$. The entries should be stored columnwise by a `double*`-array of length mn. Write `get/set`-methods for the entries of the matrix and `get`-methods for the dimensions. Moreover, write a constructor with input $m, n \in \mathbb{N}_0$, that allocates memory for a $m \times n$ matrix and initializes all entries with 0. Implement the standard constructor which generates a 0×0-matrix as well as a destructor which frees allocated memory.
Hint: Use `new` and `delete[]` for allocating and freeing memory. Also take care of the "Rule of three" (Dreierregel).

Aufgabe 10.5. Write a `Makefile` for the (programming) exercises of this sheet. It should contain:

- Compilation of all solved exercises.
- Generation of a dynamic library (e.g., for the matrix class) and an example for its usage.

Aufgabe 10.6. Explain the following C++ code. What is the functionality of `func1`, resp. `func2`? What is the output?

```
#include <iostream>
#include <vector>

using namespace std;
```

```
void func2(vector<double> &dp, int mp);
void func1(vector<double> &dp) {
   int mp = dp.size();
   func2(dp,mp);

   for (int i = mp-1; i>=1; i--){
      for(int j = 0; j<i; j++){
         if (dp[j] > dp[j+1])
            swap(dp[j], dp[j+1]); // Swap dp[j] und dp[j+1]
      }
      func2(dp,mp);
   }
}

void func2(vector<double> &dp, int mp){
   for (int k = 0; k<= mp-1; k++){
      cout << dp[k] << endl;
   }
}

int main(){
   vector<double> a(4,0);
   a[0] = 14;
   a[1] = 12;
   a[2] = 7;
   a[3] = 4;
   func1(a);
}
```

Aufgabe 10.7. What is the output of the following code? Explain why!

```
#include <iostream>
#include <string>
using namespace std;

class T1 {
   string t1;
public:
   T1(string val) { cout << "I am constructor of " << val << endl; t1=val; }
   T1() { cout << "I am constructor of default" << endl; t1="default"; }
   ~T1() { cout << "I am destructor of " << t1 << endl; }
};

int main() {
   T1 bert("bert");
   T1 bob;
   T1 def("bob");
   return 0;
}
```

Aufgabe 10.8. Explain the differences between references and pointers. Write some code which swaps the values of two variables. Implement a version which uses pointers and then implement a second version which uses references. What are the advantages of using references? What are the disadvantages?