Proseminar 2

Advanced C++

Task1

Given code throws runtime error and allocated objects are never deleted from the heap

```
int main() {
    allocate();
    throw std::runtime_error(" :) ");
    deallocate();
}
```

Solution is to wrap code with try-catch block and execute deallocation in catch part.

```
int main() {
    try{
        allocate();
        throw std::runtime_error(" :) ");
        deallocate();
    }
    catch(std::runtime_error e){        //catch exception and make sure heap object
s are deallocated
        deallocate();
    }

}
```

To make sure the potential leak is fixed, I ran command

```
valgrind --leak-check=yes ./Task2_1
```

Task 2 + 3

The description of the task says that program finishes without reaching the end. In my case, my program finished without any error. Just the file is not created. I believe the problem is in opening file with flags which actually stand in contradiction with each other:

```
file.open("test.txt",std::fstream::trunc | std::fstream::app);
```

This line says to append new text to the previous one and delete the previous text at the same time. Also, one would think that **std::fstream::app** implies **std::fstream::out** but it does not. Since I'm allowed to change nothing but main function, there's just one way how to correct the program which is **open the file once again in main - with correct flags** . Another solution would be define trunc as out in preprocessor, but I'm allowed to change only main function.

```
int main() {
    std::fstream file;
    openFile(file);
    file.open("test.txt",std::fstream::out | std::fstream::app);
    writeToFile(file);
    closeFile(file);
}
```

Task 4

I needed 3 classes - Vertex, Edge and Graph itself.

Vertex - has name and vector of edges
Edge - references 2 vertexes (cannot exist without 2 existing vertexes) and contains distance value
Graph - contains vector of vertexes

There was circular dependency between Vertex and Edge class so I forward declarated vertex in Edge class.

My implementation of graph is directed graph (one-way edges).