

N46

a)

StringInfo.h

```
#pragma once
#include <iostream>
#include <map>
#include <vector>
#include <string>
class StringInfo
{
public:
    explicit StringInfo(const std::string & str);
    void addSymbol(const char & theSymbol);
    std::vector<char> getPopularSymbols();
private:
    std::string str;
    std::map<char, int> data;
    void buildMap();
};
```

StringInfo.cpp

```
#include "StringInfo.h"
#include <algorithm>
void StringInfo::buildMap()
{
    for (const auto & i : str)
        data[i]++;
}

StringInfo::StringInfo(const std::string & theStr)
{
    str = theStr;
    buildMap();
}

void StringInfo::addSymbol(const char & theSymbol)
{
    str += theSymbol;
    data[theSymbol]++;
}

bool sortByVal(const std::pair<char, int> & a, const std::pair<char, int> & b)
{
    return (a.second > b.second);
}

std::vector<char> StringInfo::getPopularSymbols()
{
    std::vector<std::pair<char, int>> vec;
    std::map<char, int>::iterator iter;
    for (iter = data.begin(); iter != data.end(); iter++)
        vec.push_back(std::make_pair(iter->first, iter->second));
    std::sort(vec.begin(), vec.end(), sortByVal);
}
```

```

if (vec.size() > 3)
{
    std::vector<char> answer {vec[0].first, vec[1].first, vec[2].first};
    return answer;
}
else
{
    std::vector<char> answer;
    for (auto i: vec)
    {
        answer.push_back(i.first);
    }
    return answer;
}

```

B. 1.

```

std::vector<int> v {7, 8, 10, -4, 1, 5};
auto func = [&v](int x)
{
    for (std::vector<int>::iterator it = v.begin(); it != v.end(); it++)
    {
        if (*it * it - it (abs(*it - x) > (double)(x/10.0))
            std::cout << *it << " ";
    }
}

```

2.

```

auto anotherFunc = [&v](int x) {
    double for_each(v.begin(), v.end(), [&v](int t) {
        if (abs(t - x) > (double)(x/10.0))
            std::cout << t << " ";
    });
}

```