

Федеральное государственное автономное
образовательное учреждение высшего образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**
Факультет компьютерных наук

ЗАДАНИЕ №4

И снова пляшущие человечки.

по направлению подготовки Архитектура вычислительных систем
образовательная программа «Программная инженерия»

Выполнил:

Хан Се Вон, студент

БПИ-195

Преподаватель:

Легалов Александр Иванович,

Доктор технических наук,

профессор

Москва 2020

Задание

Вариант 28

И снова пляшущие человечки. Узнав о планах преступников, озвученных в задаче 33, Шерлок Холмс предложил лондонской полиции специальную машину для дешифровки сообщений злоумышленников. Реализовать многопоточное приложение, дешифрующее кодированный текст. В качестве ключа используется известная кодовая таблица, устанавливающая однозначное соответствие между каждой буквой и каким-нибудь числом. Процессом узнавания кода в решении задачи пренебречь. Каждый поток дешифрует свои кусочки текста. При решении использовать парадигму портфеля задач.

Составление программы

1. Шифрование текста для получение входных данных для программы. Для шифрования текста я использовал кодовую таблицу ASCII. (Программа предназначена для дешифрования текста на латинице)

Для шифрования был использован данный метод:

```
static string EncodeString(string text) {
    string encoded_text = "";
    for (int i = 0; i < text.length(); i++)
    {
        encoded_text += to_string((int)text[i]);
    }
    return encoded_text;
}
```

2. Получение входных данных.

Шифрованный текст программа получает путем считывания данных из файла *input.txt*.

Метод *GetDataFromFile* считывает данные и заносит зашифрованные строки в вектор.

```
//Получение данных с файла
static vector<string> GetDataFromFile(string path)
{
    ifstream fin;
    fin.open(path);
    vector<string> result;
    int numLines = GetNumOfFileLines(path);

    if (!fin.is_open())
```

```

        {
            throw invalid_argument("Error of opening the file!");
        }
        else
        {
            while (!fin.eof())
            {
                string str = "";
                getline(fin, str);
                if (str == "")
                {
                    continue;
                }
                result.push_back(str);
            }
        }
        return result;
    }
}

```

По парадигме портфеля задач я назначал количество потоков используя данную формулу:

```
ceil(packs.size() / 3)
```

В дальнейшем программа самостоятельно разбивает итерации цикла и распределяет их по потокам:

```

//Расшифрование текста из файла
static void GetDecryptedData(vector<string>
lines, int numberOfThreads)
{
    omp_set_num_threads(numberOfThreads);
#pragma omp parallel
    {
#pragma omp for
        for (int i = 0; i < lines.size(); i++)
        {
#pragma omp critical
            {
                cout << "Thread(" <<
omp_get_thread_num() << ") Index of iteration:"
<< to_string(i) << endl;
                string decrypted_text =
DecryptString(lines[i]);

result += decrypted_text + "\n";
            }
        }
    }
}

```

```

        }
    }
}

```

В цикле данного метода я вывожу полную информацию об итерации цикла (номер потока, номер итерации):

```

cout << "Thread(" << omp_get_thread_num() << "
Index of iteration:" << to_string(i) << endl;

```

Текст программы

```

#include <iostream>
#include <string>
#include <vector>
#include <fstream>
#include <thread>
#include <omp.h>

using namespace std;

static string result = "";
//Получение количества строк в файле
static int GetNumOfFileLines(string path) {
    ifstream fin;
    fin.open(path);
    string line;
    int length = 0;

    while (getline(fin, line))
        ++length;
    return length;
}
//Получение данных с файла
static vector<string> GetDataFromFile(string path)
{
    ifstream fin;
    fin.open(path);
    vector<string> result;
    int numOfLines = GetNumOfFileLines(path);

    if (!fin.is_open())
    {
        throw invalid_argument("Error of opening the file!");
    }
    else
    {

```

```

        while (!fin.eof())
        {
            string str = "";
            getline(fin, str);
            if (str == "")
            {
                continue;
            }
            result.push_back(str);
        }
    }
    return result;
}

//Шифровать текст
static string EncodeString(string text) {
    string encoded_text = "";
    for (int i = 0; i < text.length(); i++)
    {
        encoded_text += to_string((int)text[i]);
    }
    return encoded_text;
}

//Расшифровать текст
static string DecryptString(string encoded_text) {
    string decrypted_text = "";
    for (int i = 0; i < encoded_text.length() - 1; i++)
    {
        if (encoded_text[i] == '1')
        {
            char arr[] = { encoded_text[i] , encoded_text[i +
1], encoded_text[i + 2] };
            char c = stoi(arr);
            decrypted_text += c;
            i += 2;
            continue;
        }
        char arr[] = { encoded_text[i] , encoded_text[i + 1] };
        try
        {
            char c = stoi(arr);
            decrypted_text += c;
            i++;
        }
        catch (const std::exception&)
        {
            cout << arr << endl;
        }
    }
}

```

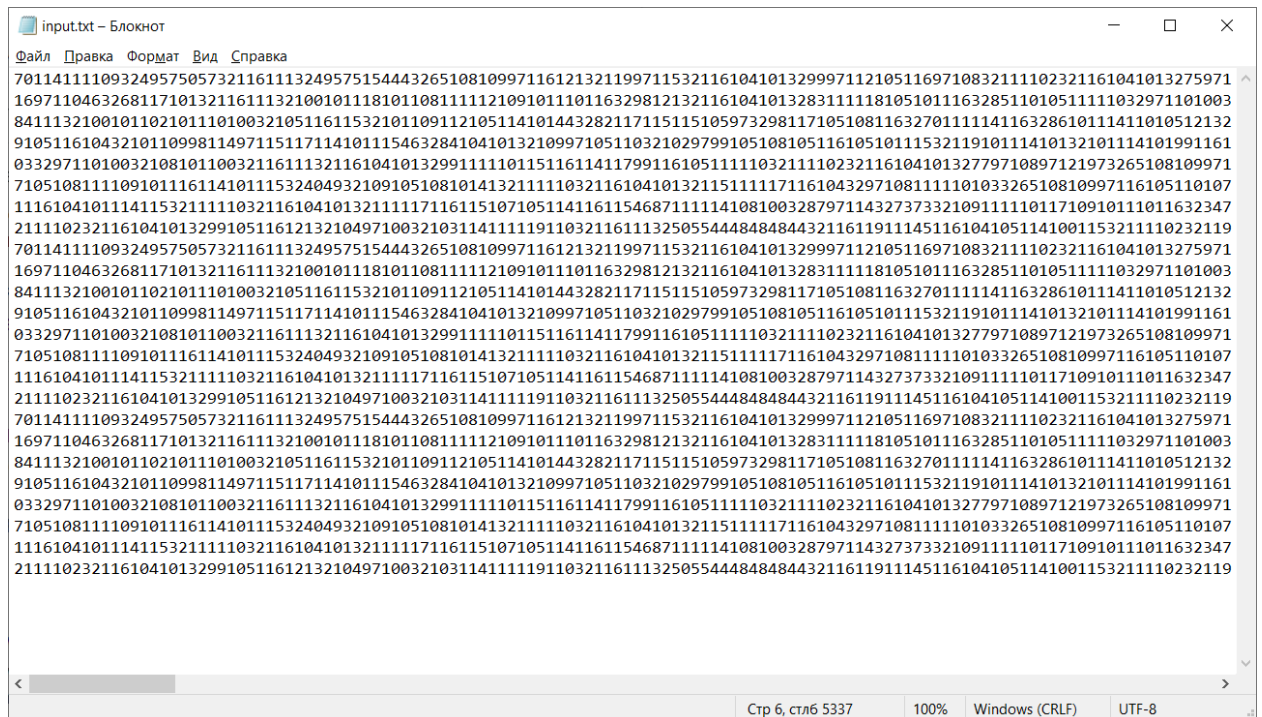
```

        return decrypted_text;
    }
    //Расшифрование текста из файла
    static void GetDecryptedData(vector<string> lines, int
    numberOfThreads)
    {
        omp_set_num_threads(numberOfThreads);
#pragma omp parallel
        {
#pragma omp for
            for (int i = 0; i < lines.size(); i++)
            {
#pragma omp critical
                {
                    cout << "Thread(" << omp_get_thread_num() <<
                    ") Index of iteration:" << to_string(i) << endl;
                    string decrypted_text =
                    DecryptString(lines[i]);
                    result += decrypted_text + "\n";
                }
            }
        }
    }
    //Получение результатов дешифрования
    static void GetResults(vector<string> packs) {
        cout << packs.size() << endl;
        GetDecryptedData(packs, ceil(packs.size() / 3));
        cout << result << endl;
    }
    int main()
    {
        try
        {
            string path = "input.txt";
            vector<string> packs = GetDataFromFile(path);
            GetResults(packs);
        }
        catch (const invalid_argument& e)
        {
            cout << "input.txt file is not found" << endl;
            return 1;
        }
        catch (const exception& e)
        {
            cout << "Incorrect encoded text" << endl;
            return 1;
        }
    }
}

```

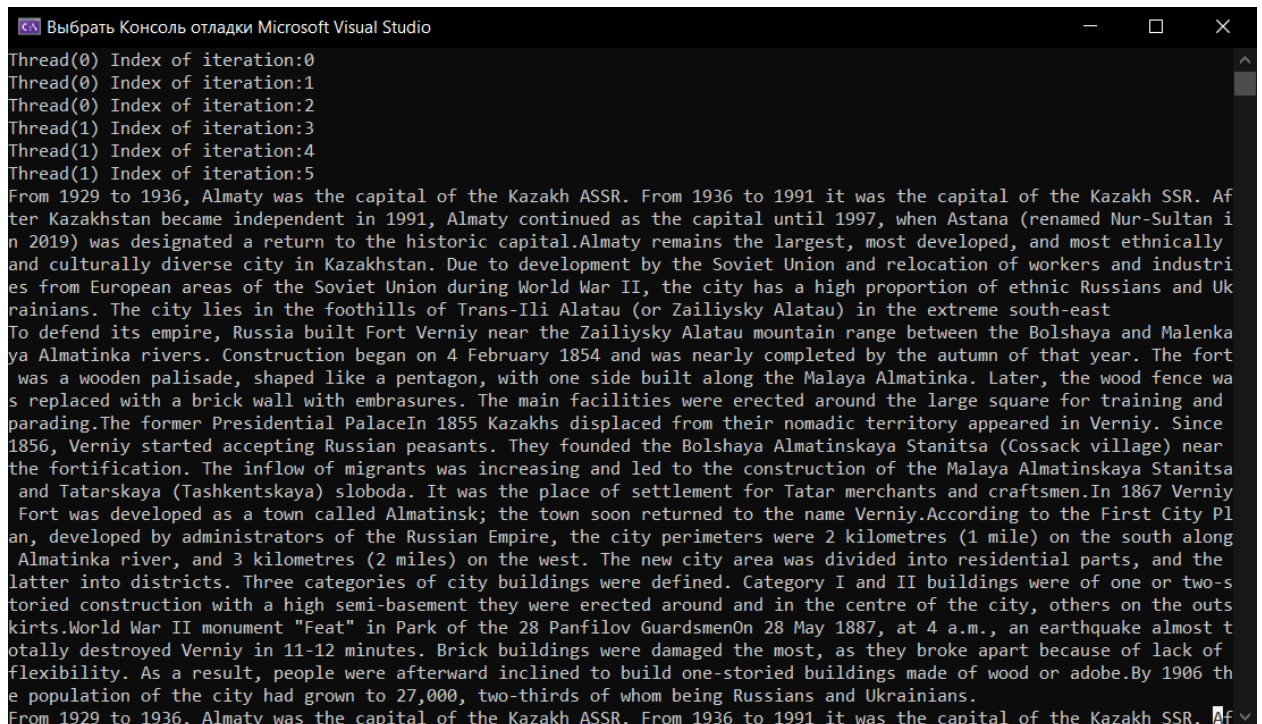
Тестирование программы

Для корректной работы программы пользователь должен поместить зашифрованный текст в файл input.txt наравне с исполняемым файлом.



```
input.txt - Блокнот
Файл  Правка  Формат  Вид  Справка
701141111093249575057321161113249575154443265108109971161213211997115321161041013299971121051169710832111102321161041013275971
169711046326811710132116111321001011181011081111121091011101163298121321161041013283111118105101116328511010511111032971101003
84111321001011021011101003210511611532101109112105114101443282117115115105973298117105108116327011114116328610111411010512132
910511610432101109981149711511711410111546328410410132109971051103210297991051081051161051011153211910111410132101114101991161
03329711010032108101100321161113211610410132991111101151161141179911610511111032111103211610410132779710897121973265108109971
7105108111109101116114101115324049321091051081014132111103211610410132115111171161043297108111110321161041013275971
1116104101114115321111032116104101321111711611510710511411611546871111141081003287971143273733210911111011710910111011632347
211110232116104101329910511612132104971003210311411119110321161113250554448484432116119111451161041051141001153211110232119
701141111093249575057321161113249575154443265108109971161213211997115321161041013299971121051169710832111102321161041013275971
169711046326811710132116111321001011181011081111121091011101163298121321161041013283111118105101116328511010511111032971101003
841113210010110210111010032105116115321011091121051141014432821171151151059732981171051081163270111114116328610111411010512132
910511610432101109981149711511711410111546328410410132109971051103210297991051081051161051011153211910111410132101114101991161
033297110100321081011003211611132116104101329911111011511611411799116105111110321111023211610410132779710897121973265108109971
7105108111109101116114101115324049321091051081014132111103211610410132115111171161043297108111110321161041013275971
1116104101114115321111032116104101321111711611510710511411611546871111141081003287971143273733210911111011710910111011632347
211110232116104101329910511612132104971003210311411119110321161113250554448484432116119111451161041051141001153211110232119
701141111093249575057321161113249575154443265108109971161213211997115321161041013299971121051169710832111102321161041013275971
169711046326811710132116111321001011181011081111121091011101163298121321161041013283111118105101116328511010511111032971101003
841113210010110210111010032105116115321011091121051141014432821171151151059732981171051081163270111114116328610111411010512132
910511610432101109981149711511711410111546328410410132109971051103210297991051081051161051011153211910111410132101114101991161
033297110100321081011003211611132116104101329911111011511611411799116105111110321111023211610410132779710897121973265108109971
7105108111109101116114101115324049321091051081014132111103211610410132115111171161043297108111110321161041013275971
1116104101114115321111032116104101321111711611510710511411611546871111141081003287971143273733210911111011710910111011632347
211110232116104101329910511612132104971003210311411119110321161113250554448484432116119111451161041051141001153211110232119
```

В результате работы в консольную программы выведется дешифрованный текст с полной информацией потоков:



```
Выбрать Консоль отладки Microsoft Visual Studio
Thread(0) Index of iteration:0
Thread(0) Index of iteration:1
Thread(0) Index of iteration:2
Thread(1) Index of iteration:3
Thread(1) Index of iteration:4
Thread(1) Index of iteration:5
From 1929 to 1936, Almaty was the capital of the Kazakh ASSR. From 1936 to 1991 it was the capital of the Kazakh SSR. Af
ter Kazakhstan became independent in 1991, Almaty continued as the capital until 1997, when Astana (renamed Nur-Sultan i
n 2019) was designated a return to the historic capital.Almaty remains the largest, most developed, and most ethnically
and culturally diverse city in Kazakhstan. Due to development by the Soviet Union and relocation of workers and industri
es from European areas of the Soviet Union during World War II, the city has a high proportion of ethnic Russians and Uk
rainians. The city lies in the foothills of Trans-Ili Alatau (or Zailiysky Alatau) in the extreme south-east
To defend its empire, Russia built Fort Verniy near the Zailiysky Alatau mountain range between the Bolshaya and Malenka
ya Almatinka rivers. Construction began on 4 February 1854 and was nearly completed by the autumn of that year. The fort
was a wooden palisade, shaped like a pentagon, with one side built along the Malaya Almatinka. Later, the wood fence wa
s replaced with a brick wall with embrasures. The main facilities were erected around the large square for training and
parading.The former Presidential PalaceIn 1855 Kazakhs displaced from their nomadic territory appeared in Verniy. Since
1856, Verniy started accepting Russian peasants. They founded the Bolshaya Almatinskaya Stanitsa (Cossack village) near
the fortification. The inflow of migrants was increasing and led to the construction of the Malaya Almatinskaya Stanitsa
and Tatarskaya (Tashkentskaya) sloboda. It was the place of settlement for Tatar merchants and craftsmen.In 1867 Verniy
Fort was developed as a town called Almatinsk; the town soon returned to the name Verniy.According to the First City Pl
an, developed by administrators of the Russian Empire, the city perimeters were 2 kilometres (1 mile) on the south along
Almatinka river, and 3 kilometres (2 miles) on the west. The new city area was divided into residential parts, and the
latter into districts. Three categories of city buildings were defined. Category I and II buildings were of one or two-s
toried construction with a high semi-basement they were erected around and in the centre of the city, others on the outs
irts.World War II monument "Feat" in Park of the 28 Panfilov GuardsmenOn 28 May 1887, at 4 a.m., an earthquake almost t
otally destroyed Verniy in 11-12 minutes. Brick buildings were damaged the most, as they broke apart because of lack of
flexibility. As a result, people were afterward inclined to build one-storied buildings made of wood or adobe.By 1906 th
e population of the city had grown to 27,000, two-thirds of whom being Russians and Ukrainians.
From 1929 to 1936, Almaty was the capital of the Kazakh ASSR. From 1936 to 1991 it was the capital of the Kazakh SSR. Af
```

При вводе некорректных данных программа выведет данное сообщение:

