Министерство науки и высшего образования РФ

ФГБОУ ВПО «Пермский государственный национальный исследовательский университет»

Кафедра прикладной математики и информатики

**Индивидуальное задание №1**

**«Числовое кольцо»**

***по дисциплине «Алгоритмизация и программирование»***

|  | **Выполнил:** |
| --- | --- |
|  | Студент 1-го курса Института КНТ  \_\_\_\_\_\_\_\_\_\_\_ Пачколин М.А.,  группа ПМИ-9,10-2022 |
|  | **Проверил:** |
|  | Преподаватель кафедры прикладной математики и информатики ПГНИУ,  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Айдаров Ю.Р. |
|  | «\_\_\_» \_\_\_\_\_\_\_\_\_ 2022 г. |

Пермь 2022

# **Постановка задачи**

# **Определение идеи алгоритма.**

Для решения задачи использовалась среда разработки Visual C++.

Для решения поставленной задачи удобно использовать тип string, позволяющий хранить в себе строку, состоящую из некоторого количества цифр.

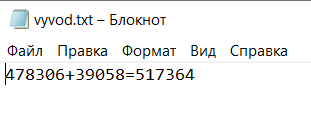
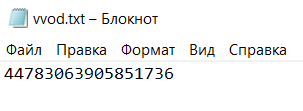
Далее будет приведен алгоритм решения:

1. Создаем функцию для проверки строки на ее состояние: является ли она полностью числовой или нет. Реализация данной функции is\_number() состоит в том, что если строка содержит в себе только цифры, то возвращает значение true, и false в противном случае.
2. Далее создаем переменную через функцию str.length(), которая обеспечивает длину заданной строки в байтах. Это фактическое количество байтов, соответствующих символам строки.
3. Далее проверяем строку на ее пригодность к дальнейшим операциям: ее количество символов, а также на условие того, что является ли строка числовой через функцию is\_number().
4. Создаем цикл для движения по кольцу с помощью сдвига исходной строки по кольцу: добавляем к строке первый символ, а далее с помощью функции erase() удаляем несколько элементов на основе позиции, указанной в первом и втором аргументах.
5. Создаем вложенные циклы для заполнения первого , второго и третьего списка будущими числами, используя при этом структуру данных list, где любой элемент знает только о предыдущем и о следующем элементах. После используется функция push\_back, которая позволяет вставлять новый элемент в конец вектора , который увеличивает размер вектора на 1.
6. Создаем функцию for\_each(), которая перебирает каждый элемент от первого до конечного из сформировавшихся списков и обрабатывает по внешней переменной [&](char n).
7. Для проверки условия A+B=C создадим условный оператор if, где значения строк Astr, Bstr, Cstr преобразуются благодаря функции atoi, которая преобразует строку string в целое значения типа int. Если условие верное, тогда мы останавливаем цикл с помощью bool`инговой переменной, меняя значение на true и выводим в выходной файл ответ.

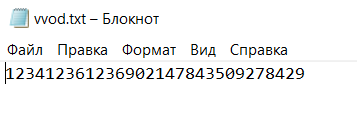
# **Тестирование программы**

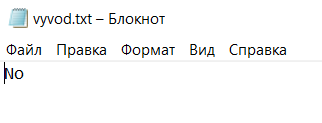
Программа была протестирована для 3-х видов входных данных, для каждого из них ниже дано описание, представлен тест и результат прохождения данного теста программой:

1. Исходные данные не имеют ошибок; на входе - число 44783063905851736



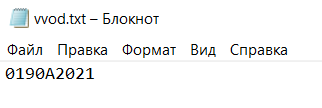
1. Исходные данные не имеют ошибок; на входе - число 123412361236902147843509278429

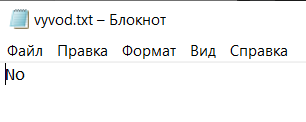




1. Исходные данные имеют ошибку: в числе присутствует буква “А”.

0190А2021





**Текст программы**

Исходный файл программы *Source.cpp* прилагается к работе в электронном виде, также приведен ниже.

В кольце записаны N цифр, составляющих по часовой стрелке 3 числа: два слагаемых и сумму. *Требуется* написать программу, которая по строке чисел и, считая ее кольцом, находит какое-нибудь решение в виде A+B=C. Все цифры числа должны входить в числа только один раз и в порядке следования в кольце. Цифр в кольце не более 1000.

*Формат входных данных:*

Входной файл содержит 1 строку, в которой без пробелов перечислены цифры кольца.

*Формат выходных данных:*

В выходной файл выводится тождество в виде <слагаемое1>+<слагаемое2>=<сумма> без пробелов внутри, или слово «No», если решения не существует.

*Пример 1 входных данных:*

01902021

*Пример 1 выходных данных:*

190+20=210

*Пример 2 входных данных:*

111111

*Пример 2 выходных данных:*

No

#include <iostream>

#include <fstream>

#include <string>

#include <list>

#include <algorithm>

using namespace std;

bool is\_digits(const string& str) {

return str.find\_first\_not\_of("0123456789") == string::npos;

}

int main()

{

setlocale(LC\_ALL, "Russian");

ifstream fin;//Чтение строки из файла

fin.open("vvod.txt");

string input\_str;

while (!fin.eof()) fin >> input\_str;

fin.close();

int input\_str\_length = input\_str.length();

if (input\_str\_length >= 3 && input\_str\_length <= 1000 && is\_digits(input\_str)) { //Проверка на условие

ofstream fout("vyvod.txt");//Открытие файла вывода

string Astr = ""; string Bstr = ""; string Cstr = "";

bool check = false; //Для остановки цикла

int cycle = 0;

while (cycle < input\_str\_length + 1 && !check) { //Двигаем кольцо

for (int j = 0; j < input\_str\_length - 2 && !check; j++) {

Astr = ""; Bstr = ""; Cstr = "";

list<char> A(1, input\_str[0]);

list<char> B(1, input\_str[j + 1]);

list<char> C;

//Заполнение первого списка

for (int i = 1; i <= j; i++) {

A.push\_back(input\_str[i]);

}

//Заполнение третьего списка оставшимеся цифрами

for (int i = j + 2; i < input\_str\_length; i++) {

C.push\_back(input\_str[i]);

}

for\_each(A.cbegin(), A.cend(), [&](char n) { Astr += n; }); // используем [&] вместо [] т.к. нам необходимо обращаться к внешней переменной

for\_each(B.cbegin(), B.cend(), [&](char n) { Bstr += n; });

for\_each(C.cbegin(), C.cend(), [&](char n) { Cstr += n; });

while (C.size() > 1 && !check) {

Astr = ""; Bstr = ""; Cstr = "";

B.push\_back(C.front()); C.pop\_front();

for\_each(A.cbegin(), A.cend(), [&](char n) { Astr += n; });

for\_each(B.cbegin(), B.cend(), [&](char n) { Bstr += n; });

for\_each(C.cbegin(), C.cend(), [&](char n) { Cstr += n; });

//Проверка условия

if (atoi(Astr.c\_str()) + atoi(Bstr.c\_str()) == atoi(Cstr.c\_str()) && Astr.front() != '0' && Bstr.front() != '0' && Cstr.front() != '0')

check = true; //Останавливаем цикл

}

ofstream fout;//Выводим ответ в файл

fout.open("vyvod.txt");

fout << Astr << "+" << Bstr << "=" << Cstr;

fout.close();

}

//Сдвигаем исходную строку по кольцу

input\_str += input\_str[0];

input\_str.erase(0, 1);

cycle++;

}

if (!check) {

ofstream fout("vyvod.txt");//Выводим в файл отрицательный ответ

fout << "No";

fout.close();

}

}

else {

ofstream fout("vyvod.txt");//Выводим в файл отрицательный ответ

fout << "No";

fout.close();

}

}