# 

ДЕПАРТАМЕНТ ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

ТОМСКОЙ ОБЛАСТИ

ОБЛАСТНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ СРЕДНЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

«ТОМСКИЙ ТЕХНИКУМ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ»

Специальность 09.02.07 «Информационные системы и программирование»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 12.2

Разработка функций с использованием данных строкового типа

|  |  |  |
| --- | --- | --- |
| Студент  «\_\_» \_\_\_\_\_\_\_\_ 2022 г. |  | А.Н. Закоурцев |
| Преподаватель  «\_\_» \_\_\_\_\_\_\_\_ 2022 г. |  | С.С. Сухушина |

Томск 2022

**Лабораторная работа № 12.2**

**Тема:** Разработка функций с использованием данных строкового типа

**ЗАДАНИЕ:** Дана строка, состоящая из букв, цифр, запятых, точек, знаков «+» и «–». Выделить подстроку, которая соответствует записи целого числа.

**Схема программы**

Начало

Enter line:

getline(cin,line)

New line: isInt(isNumber(line))

Вывод

isInt(line)

j, buff2.length(), 1

Да

Нет

hasDot = false

i = 0

buff[i] != “\n”

buff[i] != “\0”

hasDot = false

buff[j] == “.”

hasDot = true

hasDot == false

res += buff2 +”\n”

buff2= “”

i++

res

buff2 += buff[i]

Нет

Да

Да

Нет

Нет

Да

isNumber(line)

i, line.length(), 1

A

buff.size() != 0 && line[i] == '.' && isdigit(line[i+1]) && isdigit(buff.back())

A

buff += line[i]

Да

!isdigit(line[i])

Нет

B

B

buff.size() != 0 && buff.back() != '\n' && buff.back() != '.'

buff += “\n”

buff += line[i]

buff + “\n”

Нет

Да

Да

Нет

**Код программы**

#include <iostream>

#include <string>

using namespace std;

string isNumber(string line)

{

string buff;

for (int i = 0; i < line.length(); i++)

{

if (buff.size() != 0 && line[i] == '.' && isdigit(line[i+1]) && isdigit(buff.back()))

{

buff += line[i];

}

if (!isdigit(line[i]))

{

if (buff.size() != 0 && buff.back() != '\n' && buff.back() != '.')

{

buff += "\n";

}

continue;

}

buff += line[i];

}

return buff + "\n";

}

string isInt(string buff)

{

string res, buff2;

bool hasDot = false;

int i = 0, j;

while (buff[i] != '\0')

{

if (buff[i] != '\n') // если i-й элеменнт не равен переводу строки

{

buff2 += buff[i]; // в буфф2 прибавляем его

}

else // в противном случае

{

for (int j = 0; j < buff2.length(); j++) // перебираем элементы подстроки

{

hasDot = false;

if (buff2[j] == '.') // проверяем есть ли в подстроке буфф2 точка, т.е. на этом моменте уже у нас должна быть вся строка

{ // до символа перевода подстроки

hasDot = true; // если находим

break; // то выходим из фора

}

}

if (hasDot == false) // далее проверяем если нет точки

{

res += buff2 + "\n";

}

buff2 = "";

}

i++;

}

return res;

}

int main()

{

string line;

cout << "Enter line:" << endl;

getline(cin, line);

string res;

cout << "New line:\n" << isInt(isNumber(line));

//cout << "New line:\n" << isNumber(line);

system("pause");

return 0;

}

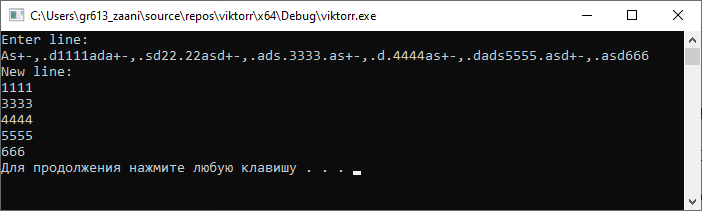
**Скриншоты работы программы**

Рисунок 1. Результат работы программы при введенной строке, которая все данные в задании символы. Программа выделяет целые записи чисел из строки

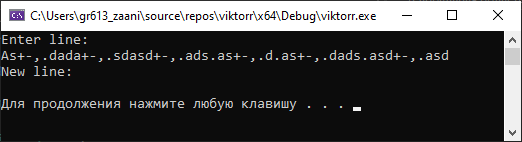
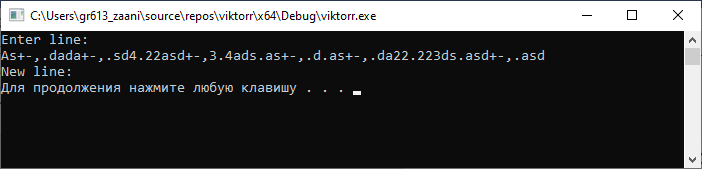
Рисунок 2. Результат работы программы при введенной строке в которой отсутствуют числа

Рисунок 3. Результат работы программы при введенной строке, которая содержит числа, которые не соответствуют записи целого числа