МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ государственное БЮДЖЕТНОЕ

образовательное учреждение

высшего образования

«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кафедра защиты информации



**Лабораторная работа №7**

*«Рекурсивные алгоритмы»*

**по дисциплине: «*Программирование*»**

|  |  |
| --- | --- |
| Выполнил:  Студент гр. «АБ-222», «АВТФ»  *Сурин Владислав Алексеевич*  «\_\_\_» \_\_\_\_\_\_ 2024г  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) | Проверил:  *Ассистент кафедры ЗИ*  *Исаев Глеб Андреевич*  «\_\_\_» \_\_\_\_\_\_ 2024г  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) |

Новосибирск 2024

**Цели и задачи работы**: изучение рекурсивного программирования, методов разработки эффективных алгоритмов.

**Задание к работе**:

* Изучить технологию использования стека при рекурсивном программировании.
* Написать рекурсивную программу решения поставленной задачи.
* Протестировать программу

**Задание № 1 (вариант 1)**

Дано число m. Требуется в последовательности цифр 1 2 3 4 … 9 расставить знаки «+» и «-» так, чтобы значением получившегося выражения было число m.

Например, m=122. Знаки расставляются как 12+34-5-6+78+9.

**Код программы (на С++)**

#include <iostream>

using namespace std;

int digit\_add(long long num, int dig)

{

while (num >= dig)

{

dig \*= 10;

}

return dig + num;

}

bool func (long long N, long long sum, long long head, long long tail)

{

if (head == 0)

{

if (sum + tail == N)

cout << tail;

return sum + tail == N;

}

if (func (N, sum + tail, head / 10, head % 10))

{

cout << "+" << tail;

}

else if (func (N, sum - tail, head / 10, head % 10))

{

cout << "-" << tail;

}

else if (func (N, sum, head / 10, digit\_add(tail, head % 10)))

{}

else

{

return false;

}

return true;

}

int main()

{

long long N, digits;

cin >> N >> digits;

func (N, 0, digits /10, digits % 10)

}

**Результат работы программы**



**Код программы (на Haskell)**

digitAdd :: Integer -> Integer -> Integer

digitAdd num dig

| num >= dig = digitAdd num (dig \* 10)

| otherwise = dig + num

func :: Integer -> Integer -> Integer -> Integer -> Bool

func n sum head tail

| head == 0 = if sum + tail == n then True else False

| func n (sum + tail) (head `div` 10) (head `mod` 10) = True

| func n (sum - tail) (head `div` 10) (head `mod` 10) = True

| func n sum (head `div` 10) (digitAdd tail (head `mod` 10)) = True

| otherwise = False

main :: IO ()

main = do

input <- getLine

let [n, digits] = map read $ words input

func n 0 (digits `div` 10) (digits `mod` 10)

**Задание № 2 (вариант 11)**

Найдите результат оценки заданного логического значения expression, представленного в виде строки. Выражение может быть либо:

* "t" эквивалентно True;
* "f" эквивалентно False;
* "!(expr)" вычисляет логическое НЕ внутреннего выражения expr;
* "&(expr1,expr2,...)" приводит к логическому И двух или более внутренних выражений expr1, expr2, ...;
* "|(expr1,expr2,...)" оценивает логическое ИЛИ двух или более внутренних выражений expr1, expr2, ...

Ограничения:

* 1 <= expression.length <= 2 \* 104
* expression[i] состоит из символов в {'(', ')', '&', '|', '!', 't', 'f', ','}.
* expression является допустимым выражением, представляющим логическое значение, как указано в описании.

Пример 1:

Ввод: выражение = "!(f)"

Вывод: правда

Пример 2:

Вход: выражение = "|(f,t)"

Выход: правда

Привет 3:

Ввод: выражение = "&(t,f)"

Вывод: ложь

**Код программы (C++)**

#include <string>

#include <iostream>

using namespace std;

size\_t end\_expr (const string& str, size\_t ind)

{

int brackets = 0;

for (; ind < str.size(); ++ind)

{

if (brackets == 0)

{

if (str[ind] == ',' or str[ind] == ')')

return ind;

}

if (str[ind] == ')')

--brackets;

if (str[ind] == '(')

++brackets;

}

return str.size();

}

bool logic\_func (const string& str, size\_t ind)

{

if (str[ind] == 't' or str[ind] == 'f')

return str[ind] == 't';

if (str[ind] == '!')

return not logic\_func(str, ind + 2);

bool is\_AND = str[ind] == '&';

ind += 2;

while (str[ind] != ')')

{

bool res = logic\_func(str, ind + 1);

if (is\_AND and not res)

{

return false;

}

if (not is\_AND and res)

{

return true;

}

ind = end\_expr(str, ind + 1);

}

return not is\_AND;

}

int main()

{

string st;

cin >> st;

cout << logic\_func(st, 0) << endl;

}

**Результаты работы программы**



**Код программы (на Haskell)**

import Data.Char

endExpr :: String -> Int -> Int

endExpr str ind = go str ind 0

where

go [] \_ \_ = length str

go (x:xs) i brackets

| brackets == 0 && (x == ',' || x == ')') = i

| x == ')' = go xs (i + 1) (brackets - 1)

| x == '(' = go xs (i + 1) (brackets + 1)

| otherwise = go xs (i + 1) brackets

logicFunc :: String -> Int -> Bool

logicFunc str ind

| isLower (str !! ind) = str !! ind == 't'

| str !! ind == '!' = not (logicFunc str (ind + 2))

| otherwise = let isAnd = str !! ind == '&'

newInd = ind + 2

in go newInd isAnd

where

go i isAnd

| str !! i == ')' = not isAnd

| otherwise = let res = logicFunc str (i + 1)

in if isAnd && not res

then False

else if not isAnd && res

then True

else go (endExpr str (i + 1)) isAnd

main :: IO ()

main = do

st <- getLine

print $ logicFunc st 0

**Вывод**

В соответствии с заданием по индивидуальному варианту были реализованы рекурсивные алгоритмы для решения определённого круга задач. Все файлы для лабораторной находятся в репозитории <https://github.com/Zl0yVlad/laba-7.git>.