**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

Лабораторная работа №5

по дисциплине «Программирование»

Рекурсия

Группа: АВТ-041

Студенты: Антонов А.

Преподаватель: Балакин В.

НОВОСИБИРСК 2021

**Задача**

Программа генерирует текст из строки, содержащей определения циклических фрагментов вида ...(Иван,Петр,Федор=Жил-был \* у самого (синего,белого,желтого= \* моря)).... Символ «\*» определяет место подстановки имени из списка в очередное повторение фрагмента. Допускается вложенность фрагментов. Полученный текст помещается в выходную строку.

**Проектирование программы**

Идея: Создание рекурсионной функции, которая вызывается, пока не избавимся от всех «\*», которая рассматривает каждый раз отдельный фрагмент

Составные части программы:

1. Массив строки и массив в котором будем рассматривать отдельные фрагменты:

char out[200], \* pout = out;

char s[80] = "(Ivan, Petr, Fedr =Jil-bil \* u samogo (Sinego,Belogo,jeltogo =\*) morya!)";

1. Выделение строки до «\*». Если в этой части будет «=», то он и то что дальше не будет учитываться, т.к цикл находится в цикле, где это учитывается:

while (\*pst != '\*') // пока не дошли до "\*"

\*pout++ = \*pst++; // pout принимает значение выделенной строки до "\*"

1. Определяем наше слово, которое стоит до «,»:

while (\*pi != ',' && \*pi != '=')

\*pout++ = \*pi++; // находим наше pout которое подставим заместо "\*"

1. Рекурсионный вызов функции, если есть ещё необработанные фрагменты:

while (\*pst != ')') // пока не дошли до конца строки

{

if (\*pst == '(') // если есть вложенный фрагмент

{

pst++;

pst = step(pst); // рекурсионный вызов, пока не избавимся от всех "\*"

}

\*pout++ = \*pst++; // новый расмматриваемый фрагмент

}

}

1. Функция main, в которой прежде чем вызвать функцию, проверяется, а надо ли её вообще вызывать:

int main()

{

char\* ps = s; // строка

for (; \*ps != 0; ps++)

if (\*ps != '(') // если нет ( нет \*

{

\*pout++ = \*ps;

}

else // если есть ( то есть \* тогда вызываем функцию

{

ps++;

ps = step(ps);

}

\*pout = 0;

printf("%s ", out);

}

Переменные:

out, s - массивы

pst, st - строка

pi - переменная-счетчик

pout - выделенная строка

ps - строка

**Текст программы с комментариями**

#include <stdlib.h>

#include <stdio.h>

char out[200], \* pout = out;

char s[80] = "(Ivan, Petr, Fedr =Jil-bil \* u samogo (Sinego,Belogo,jeltogo =\*) morya!)";

char\* step(char\* st)

{

char\* pst = st, \* pi;

while (\*pst != '=') // пока не дошли до =

pst++; // We put it on the equal symbol

pst++;

for (pi = st; \*(pi - 1) != '='; pi++)

{

if (\*pst == ')') return pst + 1;

while (\*pst != '\*') // пока не дошли до "\*"

\*pout++ = \*pst++; // pout принимает значение выделенной строки до "\*"

while (\*pi != ',' && \*pi != '=')

\*pout++ = \*pi++; // находим наше pout которое подставим заместо "\*"

pst++;

while (\*pst != ')') // пока не дошли до конца строки

{

if (\*pst == '(') // если есть вложенный фрагмент

{

pst++;

pst = step(pst); // рекурсионный вызов, пока не избавимся от всех "\*"

}

\*pout++ = \*pst++; // новый расмматриваемый фрагмент

}

}

}

int main()

{

char\* ps = s; // строка

for (; \*ps != 0; ps++)

if (\*ps != '(') // если нет ( нет \*

{

\*pout++ = \*ps;

}

else // если есть ( то есть \* тогда вызываем функцию

{

ps++;

ps = step(ps);

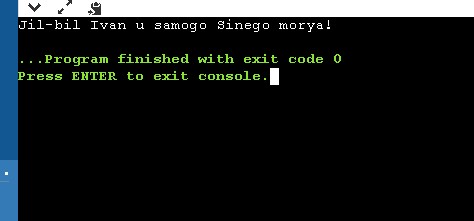
}

\*pout = 0;

printf("%s ", out);

}

**Пример работы программы**



**Вывод**  
Функция была реализована, программа работает корректно.