

**Федеральное государственное автономное образовательное
учреждение высшего образования "Национальный исследовательский
университет "Высшая школа экономики""**

**Московский институт электроники и математики им. А.Н. Тихонова
НИУ ВШЭ**

Департамент компьютерной инженерии

Курс: Алгоритмизация и программирование

Отчёт

по лабораторной работе №4

Раздел	Макс оценка	Итог. оценка
Работа программы	1	
Тесты	1	
Правильность алгоритма	3	
Ответы на вопросы	2	
Доп.задание	3	
Итого		

Студент: Андреев Глеб Андреевич

Группа: БИВ254

Вариант: №252 (1, 3, 11)

**Руководитель: Альбатша Ахмад
Мухаммад Хусайн**

Оценка: _____

Дата сдачи: _____

Оглавление

<i>Задание</i>	3
<i>Листинг</i>	4
<i>Тесты</i>	7

Задание

ЦЕЛЬ РАБОТЫ: Обработка символьных строк.

- I. (1) Выделить из каждой строки и напечатать подстроки заключенные в фигурные скобки

ПРИМЕЧАНИЕ. Пробел считать буквой. Каждую подстроку печатать с новой строки. Пустые подстроки не выделять и не печатать.

- II. (3) Среди выделенных подстрок найти подстроку содержащую цифры и круглые скобки (если таких подстрок несколько, выбирается первая из них).

ПРИМЕЧАНИЕ. Если таких подстрок несколько, выбирается первая из них.

- III.(1) Преобразовать исходную строку, которой принадлежит найденная подстрока, следующим образом: вставить три звездочки перед последней цифрой.

ЗАМЕЧАНИЕ. Все части оформить как подпрограммы в одной программе. Ввод данных, анализ существования результатов и их вывод оформить в главной программе. Если выполнение какого-либо задания невозможно, то необходимо вывести соответствующее сообщение.

Листинг

```
#include <stdio.h>
#include <string.h>
#include <ctype.h>

#define kmax 100
#define maxlen 81

// I. Выделение подстрок в фигурных скобках
int extractBraces(const char *src, char results[][][maxlen], int
*count) {
    int i = 0, n = strlen(src);
    int found = 0;

    while (i < n) {
        if (src[i] == '{') {
            int j = i + 1;
            while (j < n && src[j] != '}') j++;

            if (j < n && j > i + 1) {
                int len = j - i - 1;
                strncpy(results[*count], src + i + 1, len);
                results[*count][len] = '\0';
                (*count)++;
                found = 1;
            }
            i = j;
        }
        i++;
    }
    return found;
}

// II. Проверка, содержит ли подстрока цифры и круглые скобки
int hasDigitsAndParentheses(const char *s) {
    int digit = 0, brace = 0;
    for (int i = 0; s[i]; i++) {
        if (isdigit(s[i])) digit = 1;
        if (s[i] == '(' || s[i] == ')') brace = 1;
    }
    if (digit && brace)
        return 1;
    return 0;
}

// III. Вставить "****" перед последней цифрой строки
int insertStars(char *s) {
    int pos = -1;
    for (int i = 0; s[i]; i++)
        if (isdigit(s[i]))
            pos = i;
```

```

if (pos == -1) return 0;

char temp[maxlen];
strcpy(temp, s + pos);

s[pos] = '\0';
strcat(s, "***");
strcat(s, temp);

return 1;
}

int main() {
    int k;
    char check;
    printf("Lab work #4\n");
    do {
        printf("Enter the number of strings (from 1 to %d): ", kmax);
        scanf("%d", &k);
        check = getchar();
        if (check != '\n' || k <= 0) {
            printf("Invalid input (n must be a natural number).\n");
            k = 0;
            while (check != '\n' && check != EOF) {
                check = getchar();
            }
        }
    } while (k <= 0 || k > kmax);

    char lines[kmax][maxlen];
    printf("Insert %d lines:\n", k);
    for (int i = 0; i < k; i++) {
        fgets(lines[i], maxlen, stdin);
        lines[i][strcspn(lines[i], "\n")] = '\0';
    }

    char substrings[kmax * 100][maxlen];
    int total = 0;
    int owner[kmax * 100];

    // I. Извлекаем подстроки
    for (int i = 0; i < k; i++) {
        int startCount = total;
        extractBraces(lines[i], substrings, &total);
        for (int j = startCount; j < total; j++)
            owner[j] = i;
    }

    if (total == 0) {

```

```

        printf("There are no non-empty substrings in curly
braces.\n");
        return 0;
    }

    printf("\nExtracted substrings:\n");
    for (int i = 0; i < total; i++)
        printf("%s\n", substrings[i]);

    // II. Находим первую подстроку, содержащую цифры и круглые
    скобки
    int target = -1;
    for (int i = 0; i < total; i++) {
        if (hasDigitsAndParentheses(substrings[i])) {
            target = i;
            break;
        }
    }

    if (target == -1) {
        printf("\nThere is no substring containing numbers and
parentheses.\n");
        return 0;
    }
    printf("\nSubstring was found: %s", substrings[target]);

    // III. Преобразуем исходную строку
    int lineIndex = owner[target];

    if (!insertStars(lines[lineIndex])) {
        printf("\nThere are no numbers to insert in the source
string.\n");
        return 0;
    }

    printf("\nConverted string:\n%s\n", lines[lineIndex]);

    return 0;
}

```

Тесты

№	Исходные данные	Результаты
1	k = 2 lines = ["dfs{1ed(1)}", "d{}asd{}134(90)"]	Extracted substrings: 1ed(1) Substring was found: 1ed(1) Converted string: dfs{1ed(***)}
2	k = 1 lines = ["flksdf{}"]	There are no non-empty substrings in curly braces.
3	k = 3 lines = ["dsa{asda}", "da12314//{3456}", "dsa09{dsf()}fds"]	Extracted substrings: asda 3456 dsf() There is no substring containing numbers and parentheses.