**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: Информатика и вычислительная техника**

**ОТЧЕТ**

**по лабораторной работе №3**

**Студент: Зайцев Сергей Алексеевич**

**Группа: БИВ182**

**Вариант: №6,7,10**

**Руководитель: Ерохина Елена Альфредовна**

**Оценка:\_\_\_\_\_\_\_\_\_\_**

|  |  |  |  |
| --- | --- | --- | --- |
| **Раздел** | **Max оценка** | **Задание 1** | **Задание 2** |
| **Работа программы** | **1** |  |  |
| **Правильность алгоритма** | **3** |  |  |
| **Тесты** | **1** |  |  |
| **Вопросы** | **2** |  |  |
| **Доп. задание** | **3** |  |  |

**МОСКВА 2019**

Оглавление

[Задание 2](#_Toc506212619)

[Листинг программы 2.1 3](#_Toc506212620)

[Листинг программы 2.2 4](#_Toc506212621)

[Тесты 5](#_Toc506212622)

# Задание

6. Заключенные в скобки комментария /\* и \*/.

7. Содержащую максимальное число строчных латинских букв.

10. Заменить первые 10 заглавных букв латинского алфавита на соответствующие цифры ( А на 0, В на 1, С на 2 … J на 9). ПРИМЕЧАНИЕ. Если вариант содержит несовместимые задания, следует обратиться к преподавателю для изменения набора заданий.

# Листинг программы

#include <stdio.h>

#include <string.h>

#define MAXSTR 10

#define MAXLEN 81

void part1(int n, char str[][MAXLEN], char str1[][MAXLEN], int \*n1, int num[])

{

char \*s, \*s1;

\*n1 = 0;

for (int i = 0; i < n; i++)

{

s = str[i];

s1 = s;

for (; \*s; s++)

{

if (\*(s-1) == '/' && \*s == '\*')

{

s1 = ++s;

for (; \*s && !(\*s == '\*' && \*(s+1) == '/'); s++);

if (s1 != s && \*s == '\*' && \*(s+1) == '/')

{

num[\*n1] = i;

strncpy(str1[\*n1], s1, s - s1);

str1[\*n1][s-s1] = '\0';

(\*n1)++;

}

}

}

}

}

void part2(char str1[][MAXLEN], int n1, int \*maxstr)

{

int max = 0;

char \*s;

for (int i = 0; i < n1; i++) {

s = str1[i];

int k=0;

while (\*s)

{

if ((\*s >= 'a' && \*s <= 'z') || (\*s >= 'A' && \*s <= 'Z'))

{

++k;

//printf("%s", \*s);

//printf("max = %d", max);

//printf("k = %d", k);

//printf("\n");

max = k;

\*maxstr = i;

}

s++;

}

}

}

void part3(int num[], int maxstr, char str[][MAXLEN], char str1[][MAXLEN], int n, char \*\*flag)

{

char \*s; int i;

for (i = 0; \*flag == NULL && i < n; ++i)

\*flag = strstr(str[i], str1[maxstr]);

if (\*\*flag)

for (s = str[i-1]; \*s; ++s)

switch (\*s)

{

case 'A': \*s = '0'; break;

case 'B': \*s = '1'; break;

case 'C': \*s = '2'; break;

case 'D': \*s = '3'; break;

case 'E': \*s = '4'; break;

case 'F': \*s = '5'; break;

case 'G': \*s = '6'; break;

case 'H': \*s = '7'; break;

case 'I': \*s = '8'; break;

case 'J': \*s = '9'; break;

}

}

int main()

{

char str[MAXSTR][MAXLEN], str1[MAXSTR+20][MAXLEN], \*flag = NULL;

int num[MAXSTR], n1, n = 0, maxstr = 0;

puts("Input strings:");

while (\*gets(str[n]) && n < MAXSTR) n++;

part1(n, str, str1, &n1, num);

if (n1 == 0) puts ("No substrings.");

else

{

puts("Substrings:");

for (int i = 0; i < n1; i++)

puts(str1[i]);

}

part2(str1, n1, &maxstr);

if (maxstr != 0)

{

puts("The substring which starts with maximum number of letters has been found:");

puts(str1[maxstr]);

part3(num, maxstr, str, str1, n, &flag);

puts("The matrix after executing part3:");

for (int i = 0; i < n; ++i)

puts(str[i]);

}

else puts("The substring hasn't been found.");

}

# Тесты

