

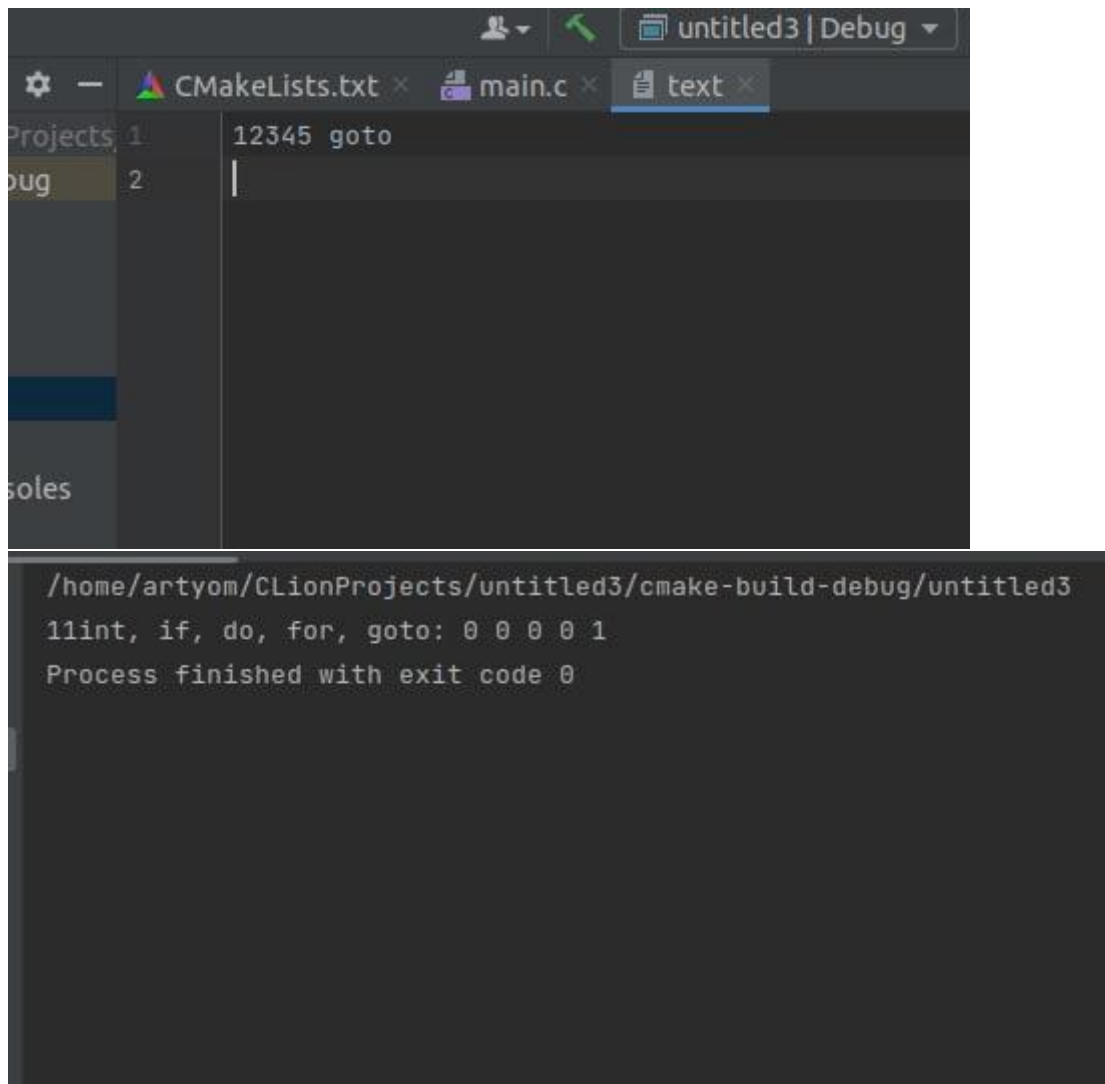
ИДЗ №2 (Отчет на 5)

Воробьев Артём Александрович БПИ214 Вариант №27

Условие задания: Разработать программу, которая определяет частоту встречаемости (сколько раз встретилось в тексте) пяти ключевых слов языка программирования C, в произвольной ASCII-строке. Ключевые слова не должны являться частью идентификаторов. Пять искоемых ключевых слов выбрать по своему усмотрению. Тестировать можно на файлах программ.

- 1) Тесты для проверки работы программ на Си и Ассемблере: (программа выводит длину строки, ключевые слова, выбранные мной (int, if, do, for, goto), и соответственное кол-во вхождений в строку каждого их них.

1.



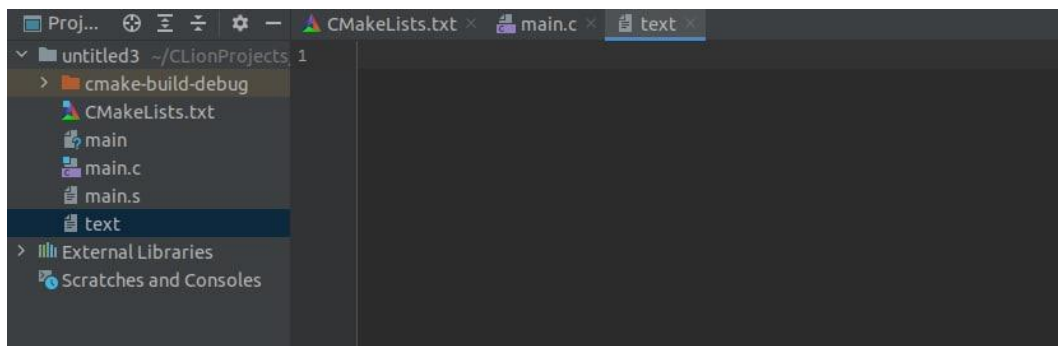
The screenshot shows the CLion IDE interface. The top toolbar includes icons for settings, a file explorer, and a debug button. The file explorer shows a project named 'untitled3' with a 'Debug' configuration. The main editor displays a C program in 'main.c' and a text file 'text'. The 'text' file contains the string '12345 goto'. The output console at the bottom shows the program's execution results.

```
12345 goto
```

```
/home/artiom/CLionProjects/untitled3/cmake-build-debug/untitled3
11int, if, do, for, goto: 0 0 0 0 1
Process finished with exit code 0
```

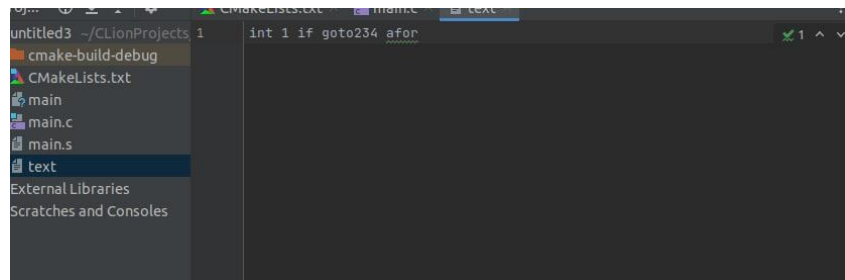
```
artyom@artyom-GL703VM:~/CLionProjects/untitled3$ ./main
11int, if, do, for, goto: 0 0 0 0 1 artyom@artyom-GL703VM:~/CLionProjects/untitled3$
```

2.



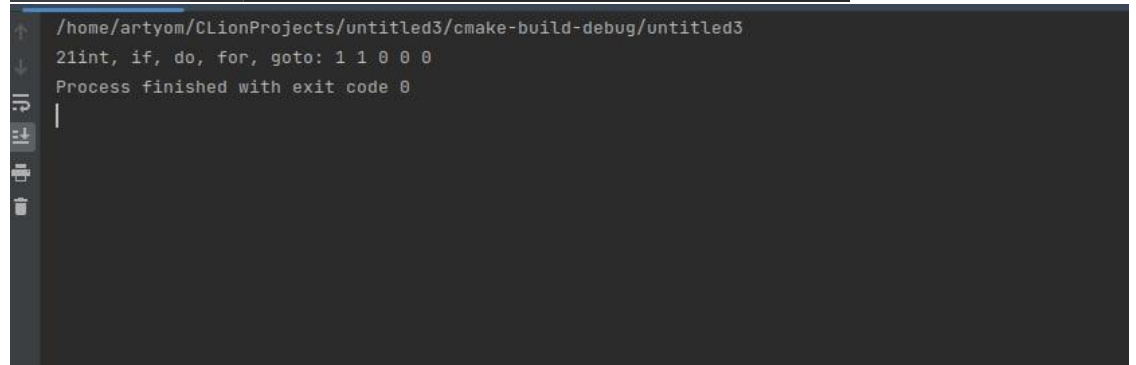
```
artyom@artyom-GL703VM:~/CLionProjects/untitled3$ ./main
11int, if, do, for, goto: 0 0 0 0 1 artyom@artyom-GL703VM:~/CLionProjects/untitled3$ ./main
0Пустой файл artyom@artyom-GL703VM:~/CLionProjects/untitled3$
```

3.



```
untitled3 ~/CLionProjects/1
cmake-build-debug
CMakeLists.txt
main
main.c
main.s
text
External Libraries
Scratches and Consoles
```

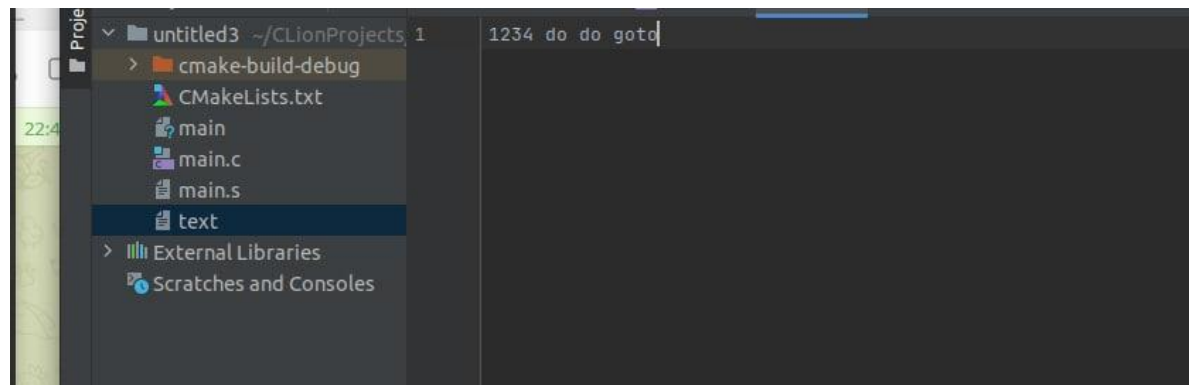
```
int 1 if goto234 afor
```



```
/home/artiom/CLionProjects/untitled3/cmake-build-debug/untitled3
21int, if, do, for, goto: 1 1 0 0 0
Process finished with exit code 0
```

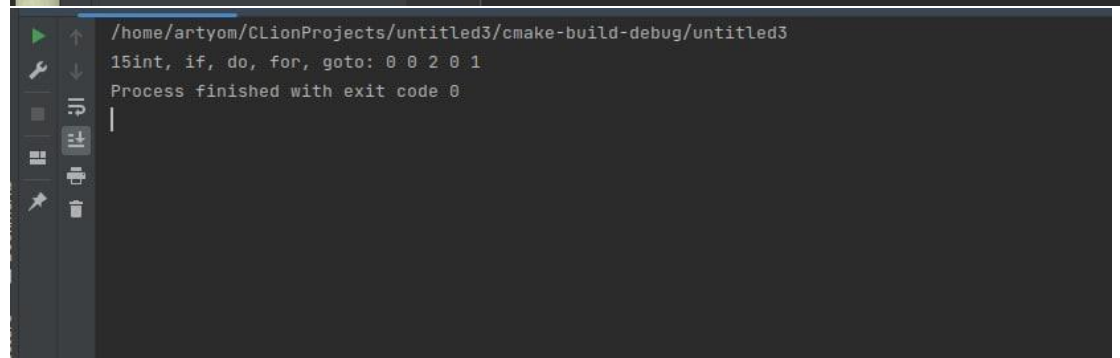
```
11int, if, do, for, goto: 0 0 0 0 1 artiom@artiom-GL703VM:~/CLionProjects/untitled3$ ./main
0Пустой файл artiom@artiom-GL703VM:~/CLionProjects/untitled3$ ./main
21int, if, do, for, goto: 1 1 0 0 0 artiom@artiom-GL703VM:~/CLionProjects/untitled3$
```

4.



```
untitled3 ~/CLionProjects/1
cmake-build-debug
CMakeLists.txt
main
main.c
main.s
text
External Libraries
Scratches and Consoles
```

```
1234 do do goto
```



```
/home/artiom/CLionProjects/untitled3/cmake-build-debug/untitled3
15int, if, do, for, goto: 0 0 2 0 1
Process finished with exit code 0
```

```
artyom@artyom-GL703VM:~/CLionProjects/untitled3$ ./main
15int, if, do, for, goto: 0 0 2 0 1 artyom@artyom-GL703VM:~/CLionProjects/untitl
ed3$
```

5.

The screenshot shows the CLion IDE interface. The top editor pane displays a C program with the following code:

```
1234 do do goto goto goto int aaaaiffodfo goto
```

The bottom pane shows the output of the program's execution:

```
/home/artyom/CLionProjects/untitled3/cmake-build-debug/untitled3
46int, if, do, for, goto: 1 0 2 0 4
Process finished with exit code 0
```

Below the IDE screenshot, a terminal window shows the command execution and output:

```
ed3$ ./main
46int, if, do, for, goto: 1 0 2 0 4 artyom@artyom-GL703VM:~/CLionProjects/untitl
ed3$
```

2) Информация, подтверждающая выполнение задания в соответствии требованиям на предполагаемую оценку.

Комментарии:

1.

```

.type    Len, @function
Len:
    ## функция int Len
    push    rbp
    mov     rbp, rsp
    sub     rsp, 32
    mov     QWORD PTR -24[rbp], rdi ## rdi - это строка s
    mov     rax, QWORD PTR -24[rbp]
    mov     rdi, rax ## через rdi передаем строку
    call    strlen@PLT ## получаем ее длину
    mov     DWORD PTR -4[rbp], eax ## присваиваем x длину строки s
    mov     eax, DWORD PTR -4[rbp] ## возвращаем x
    leave
    ret
.size    Len, .-Len
.section     .rodata
.LC0:
    .string "len"

int Len(char *s){
    int x;
    x = strlen(s);
    return x;
}

```

2.

```

.L4:
    mov     rdx, QWORD PTR -10056[rbp]
    lea     rax, -10016[rbp]
    mov     esi, 10000
    mov     rdi, rax
    call    fgets@PLT ## считываем строку
    test    rax, rax ## сравнение строки с 0
    jne     .L4 ## считываем еще,если не 0
    lea     rax, -10016[rbp]
    mov     rdi, rax ## передаем через rdi строку s
    call    Len ## вызываем функцию Len
    mov     DWORD PTR -10060[rbp], eax ## eax - длина строки (x)
    mov     eax, DWORD PTR -10060[rbp]
    mov     esi, eax
    lea     rax, .LC2[rip]
    mov     rdi, rax
    mov     eax, 0
    call    printf@PLT ## выводим длину строки
    cmp     DWORD PTR -10060[rbp], 0 ## сравниваем с 0 длину, если 0 прыгаем на .L5
    je      .L5
    mov     DWORD PTR -10068[rbp], 0
    jmp     .L6

```

```

while (fgets(s, 10000, f) != NULL);
int l;
l = Len(s);
printf("%d", l);
if (l != 0) {

```

.L22:

```
mov     eax, DWORD PTR -10068[rbp]
cdqe
movzx   eax, BYTE PTR -10016[rbp+rax]
cmp     al, 105 ## s[i] == 'i'
jne     .L7

mov     eax, DWORD PTR -10060[rbp]
sub     eax, 2
cmp     DWORD PTR -10068[rbp], eax ## i <= 1 - 3
jge     .L7

cmp     DWORD PTR -10068[rbp], 0 ## (i == 0)
je      .L8

mov     eax, DWORD PTR -10068[rbp] ## (s[i - 1] == ' ')
sub     eax, 1
cdqe
movzx   eax, BYTE PTR -10016[rbp+rax]
cmp     al, 32
jne     .L7
```

.L8:

```
mov     eax, DWORD PTR -10068[rbp] ## (s[i + 1] == 'n')
add     eax, 1
cdqe
movzx   eax, BYTE PTR -10016[rbp+rax]
cmp     al, 110
jne     .L7

mov     eax, DWORD PTR -10068[rbp] ## (s[i + 2] == 't')
add     eax, 2
cdqe
movzx   eax, BYTE PTR -10016[rbp+rax]
cmp     al, 116
jne     .L7

mov     eax, DWORD PTR -10060[rbp] ## (i == 1 - 3)
sub     eax, 3
cmp     DWORD PTR -10068[rbp], eax
je      .L9

mov     eax, DWORD PTR -10068[rbp] ## (s[i + 3] == ' ')
..
```

```

        cmp     al, 32
        jne     .L7
.L9:
        mov     eax, DWORD PTR -10048[rbp]
        ## X[0] += 1
        add     eax, 1
        mov     DWORD PTR -10048[rbp], eax
        jmp     .L10

if (s[i] == 'i' && (i <= 1 - 3) && ((i == 0) || (s[i - 1] == ' ')) && (s[i + 1] == 'n') &&
(s[i + 2] == 't') &&
((i == 1 - 3) || (s[i + 3] == ' '))) {
    X[0] += 1; // int

```

4.

```

.L10:
        add     DWORD PTR -10068[rbp], 1 ## i+= 1
.L6:
        mov     eax, DWORD PTR -10068[rbp]
        cmp     eax, DWORD PTR -10060[rbp] ## сравниваем i, 1
        jl      .L22 ## если меньше, повторяем цикл
        lea     rax, .LC3[rip]
        mov     rdi, rax
        mov     eax, 0
        call    printf@PLT ## выводим "int, if, do, for, goto: "
        mov     DWORD PTR -10064[rbp], 0
        jmp     .L23 ## идем выводить массив

for (int i = 0; i < 1; i++) {
}
printf("int, if, do, for, goto: ");

```

Остальные комментарии добавлять в отчет не стал, но они есть в файле main.s. Лишних макросов в ассемблерной программе не оказалось.

- 3) Была использована функция `int Len(char *s)` с передачей данных через параметры, а также локальная переменная `int x;` (см. п.1 в Комментариях)

Программа скомпилирована с помощью

`gcc -O0 -Wall -masm=intel -S -fno-asynchronous-unwind-tables -fcf-protection=none main.c`

и

`gcc main.s -o main`