1. Программа запрашивает имя и фамилию пользователя, после чего приветствует его.

flat assembler 1.73.25

File Edit Search Run Options Help

```
format PE console

entry start

include 'win32a.inc'

section '.data' data readable writable
        strN db 'Please, enter your name: ', 0
        strS db 'Please, enter your surname: ', 0
        strH db 'Hi, %s %s, nice to see you again!', 0

        spaceStrN db '%s', 0
        spaceStrS db '%s', 0

        N rd 2
        S rd 2

        NULL = 0


section '.code' code readable executable
        start:
                push strN
                call [printf]

                push N
                push spaceStrN
                call [scanf]

                push strS
                call [printf]

                push S
                push spaceStrS
                call [scanf]

                push S
                push N
                push strH
                call [printf]

                call [getch]

                push NULL
                call [ExitProcess]
```

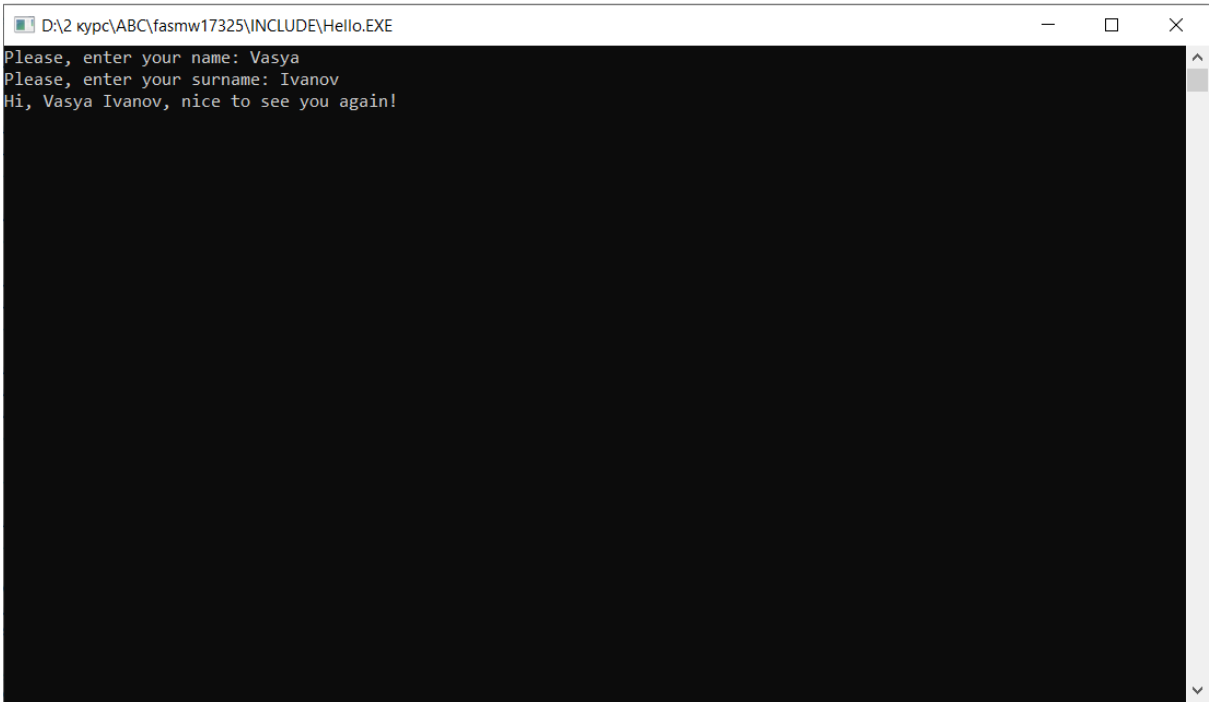
Hello.ASM

1,1

```
                push NULL
                call [ExitProcess]

section '.idata' import data readable
        library kernel, 'kernel32.dll',\
                msvcrt, 'msvcrt.dll'

        import kernel,\
               ExitProcess, 'ExitProcess'

        import msvcrt,\
               printf, 'printf',\
               scanf, 'scanf',\
               getch, '_getch'
```

Hello.ASM

1,1


D:\2 курс\ABC\fasmw17325\INCLUDE\Hello.EXE
Please, enter your name: Vasya
Please, enter your surname: Ivanov
Hi, Vasya Ivanov, nice to see you again!

2. Программа находит площадь квадрата по заданной стороне.

flat assembler 1.73.25

File Edit Search Run Options Help

```
format PE console

entry start

include 'win32a.inc'

section '.data' data readable writable
        strS db 'To calculate square of square enter the size of the side: ', 0
        strRes db 'Square = %d', 0

        size rd 1
        formatSize db '%d', 0

        NULL = 0


section '.code' code readable executable
        start:
              push strS
              call [printf]

              push size
              push formatSize
              call [scanf]

                 mov ecx, [size]
                 imul ecx, [size]

                 push ecx
                 push strRes
                 call [printf]


              call [getch]

              push NULL
              call [ExitProcess]


section '.idata' import data readable
        library kernel, 'kernel32.dll',\
                msvcrt, 'msvcrt.dll'

        import kernel,\
               ExitProcess, 'ExitProcess'
```

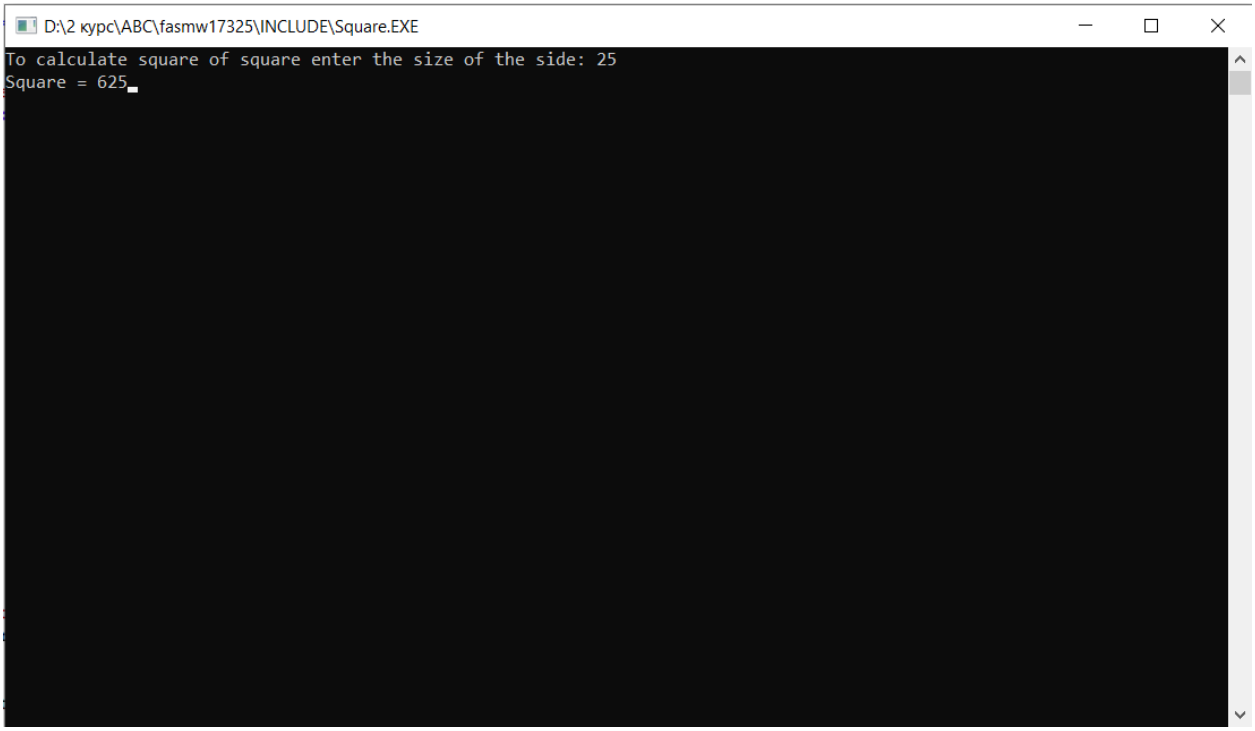
Square.ASM

1,1

```
              push NULL
              call [ExitProcess]


section '.idata' import data readable
        library kernel, 'kernel32.dll',\
                msvcrt, 'msvcrt.dll'

        import kernel,\
               ExitProcess, 'ExitProcess'

        import msvcrt,\
               printf, 'printf',\
               scanf, 'scanf',\
               getch, '_getch'
```

Square.ASM

1,1


D:\2 курс\ABC\fasmw17325\INCLUDE\Square.EXE
To calculate square of square enter the size of the side: 25
Square = 625

3. Программа вычисляет установлен ли флаг или нет, по заданным значениям.

flat assembler 1.73.25

File Edit Search Run Options Help

```
format PE console

entry Start

include 'win32a.inc'

section '.data' data readable writable
        resStr db 'Result: %d', 0
        str1 db 'ZF = 1, op result is FALSE', 0
        str2 db 'ZF = 0, op result is TRUE', 0

        A dw ?
        B dw ?
        C dw ?

        NULL = 0

section '.code' code readable executable
        Start:
                mov eax, 1010b
                test eax, 101b

                jz ifZFTrue

                push str2
                call [printf]

                jmp finish

                ifZFTrue:
                        push str1
                        call [printf]


                finish:
                        call [getch]

                        push NULL
                        call [ExitProcess]


         call [getch]

                 push NULL
                 call [ExitProcess]
```

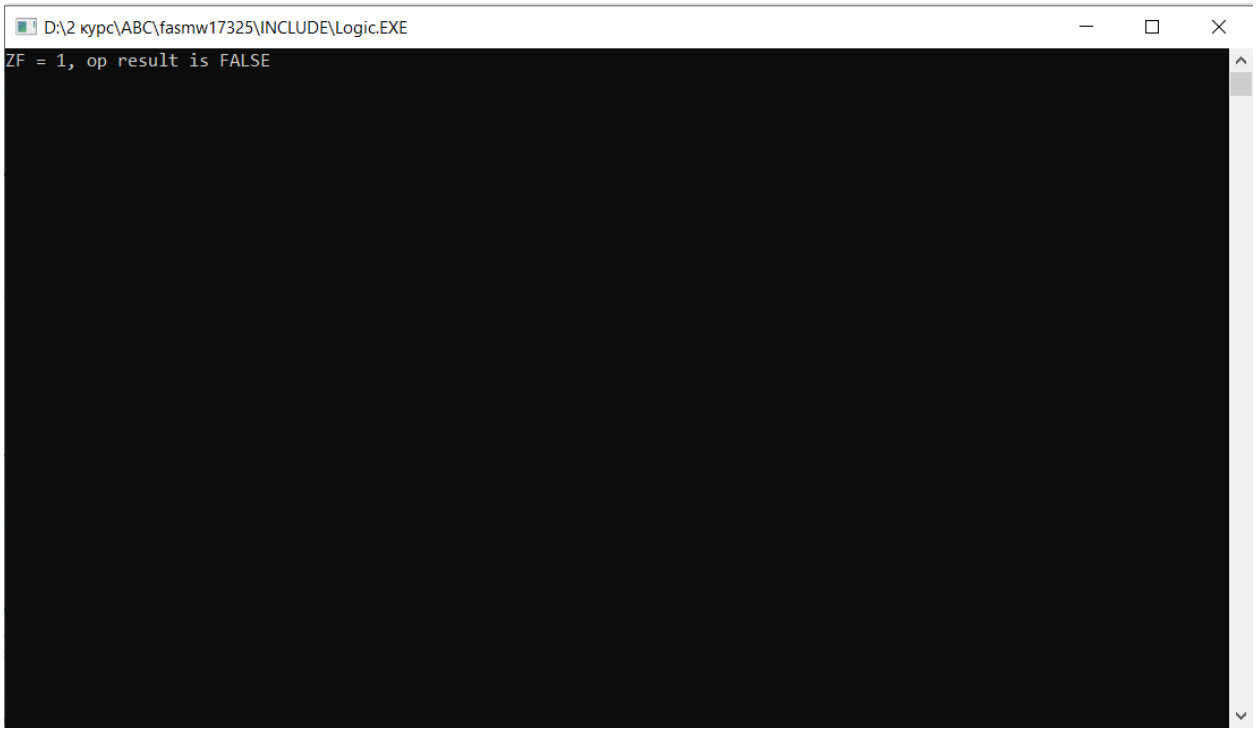
Logic.ASM

1,1

```
section '.idata' import data readable
        library kernel, 'kernel32.dll',\
                msvcrt, 'msvcrt.dll'

        import kernel,\
               ExitProcess, 'ExitProcess'

        import msvcrt,\
               printf, 'printf',\
               scanf, 'scanf',\
               getch, '_getch'
```

Logic.ASM

1,1


D:\2 курс\ABC\fasmw17325\INCLUDE\Logic.EXE
ZF = 1, op result is FALSE

4. Программа спрашивает, что хочет вычислить пользователь, затем запрашивает некоторые значения и выдает ответ.

flat assembler 1.73.25

File Edit Search Run Options Help

```
format PE console

entry Start

include 'win32a.inc'

section '.data' data readable writable
         strE db 'Please, write what you want to find: speed - 1 or distance - 2: ', 0
         str1 db 'Please enter distance: ', 0
         str2 db 'Please enter time: ', 0
         str3 db 'Please enter speed: ', 0
         strRes db 'Result = %d', 0

         spaceStr db '%d', 0

         D dd ?
         T dd ?
         S dd ?

         NULL = 0


section '.code' code readable executable
        Start:
                push strE
                call [printf]

                call [getch]

                cmp eax, 49
                jne notSpeed
                    push str1
                    call [printf]

                    push D
                    push spaceStr
                    call [scanf]

                    push str2
                    call [printf]

                    push T
                    push spaceStr
                    call [scanf]
```

Speed.ASM

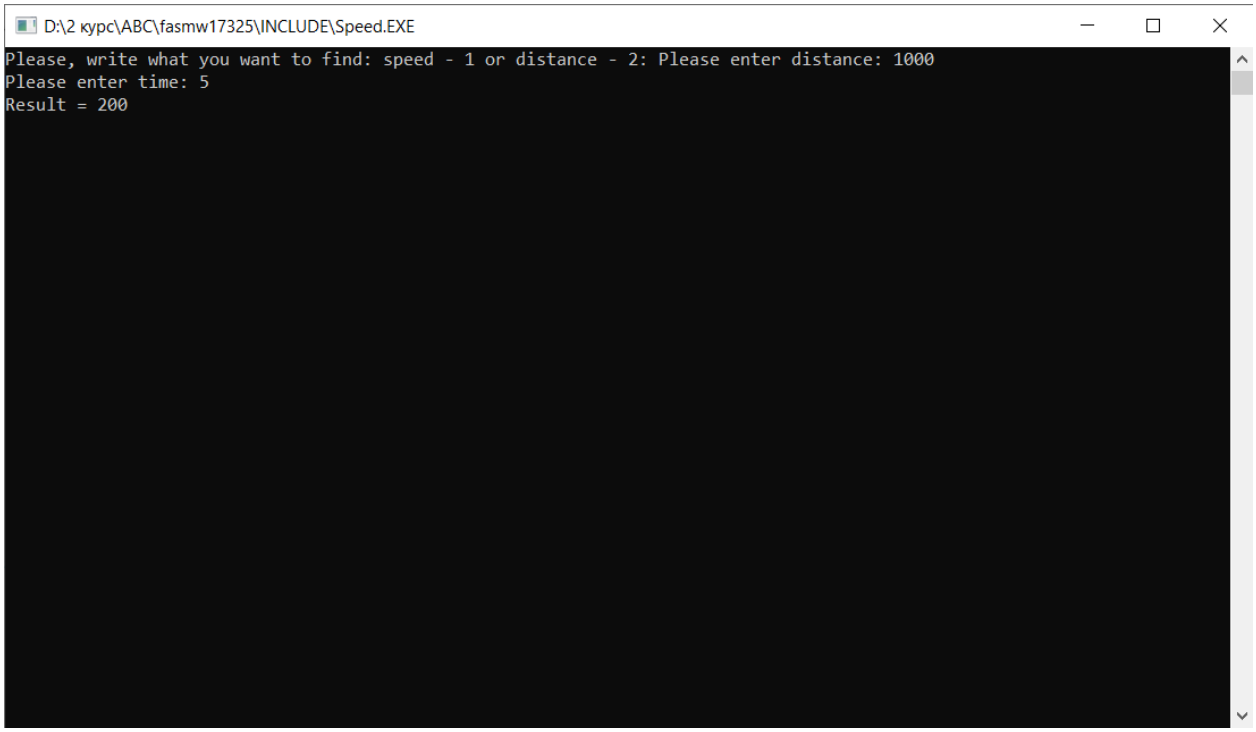
1,1

```
                mov eax, [D]
                mov ecx, [T]
                mov edx, 0

                div ecx

                push eax
                push strRes
                call [printf]

                jmp finish
            notSpeed:


            push str2
            call [printf]

            push T
            push spaceStr
            call [scanf]

            push str3
            call [printf]

            push S
            push spaceStr
            call [scanf]

            mov eax, [T]
            mul [S]

            push eax
            push strRes
            call [printf]

        finish:

        call [getch]
```

Speed.ASM

1,1

```
        finish:

        call [getch]

        push NULL
        call [ExitProcess]



section '.idata' import data readable
        library kernel, 'kernel32.dll',\
                msvcrt, 'msvcrt.dll'

        import kernel,\
               ExitProcess, 'ExitProcess'

        import msvcrt,\
               printf, 'printf',\
               scanf, 'scanf',\
               getch, '_getch'
```


D:\2 курс\ABC\fasmw17325\INCLUDE\Speed.EXE
Please, write what you want to find: speed - 1 or distance - 2: Please enter distance: 1000
Please enter time: 5
Result = 200

5. Программа запрашивает 2 числа у пользователя и операцию, которую он хочет вычислить (упрощенный калькулятор).

flat assembler 1.73.25

File Edit Search Run Options Help

```
format PE console

entry Start

include 'win32a.inc'

section '.data' data readable writable
        strA db 'Enter A: ', 0
        strB db 'Enter B: ', 0
        strOp db 'Enter operation: ', 0

        resStr db 'Result: %d', 0
        resMod db '/%d', 0

        spaceStr db ' %d', 0
        emptyStr db '%d', 0

        infinity db 'infinity', 0
        point db ',', 0

        A dd ?
        B dd ?
        C dd ?

        NULL = 0

section '.code' code readable executable
        Start:
                push strA
                call [printf]

                push A
                push spaceStr
                call [scanf]

                push strB
                call [printf]

                push B
                push spaceStr
                call [scanf]

                push strOp
                call [printf]
```

```
call [getch]

cmp eax, 43       ; + Сравниваем полученное значение для операции со значением операции по таблице
jne notAdd        ;Если получили значение 0, то переходим к метке notAdd
    mov ecx, [A]
    add ecx, [B]

    push ecx
    push resStr
    call [printf]

    jmp finish     ;Если операция прошла, то завершаем программу
 notAdd:

 cmp eax, 45       ; -
 jne notSub
     mov ecx, [A]
     sub ecx, [B]

     push ecx
     push resStr
     call [printf]

     jmp finish
 notSub:

 cmp eax, 42
 jne notMul
     mov ecx, [A]
     imul ecx, [B]

     push ecx
     push resStr
     call [printf]

     jmp finish
 notMul:

 cmp eax, 37
 jne notMod
```

Calculator.ASM

44,31

```
cmp eax, 37
jne notMod
    mov eax, [A]
    mov ecx, [B]
    mov edx, 0

    cmp [B], 0
    jne notNullDiv
        push infinity
        call [printf]
    notNullDiv:

    div ecx
    mov [C], edx

    push eax
    push resStr
    call [printf]

    push [C]
    push spaceStr
    call [printf]

    push [B]
    push resMod
    call [printf]

    jmp finish
notMod:

cmp eax, 47
jne notDiv
    mov eax, [A]
    mov ecx, [B]
    mov edx, 0

    cmp [B], 0
    jne notNullDiv1
        push infinity
        call [printf]

        jmp finish
    notNullDiv1:
```

```
        div ecx
        mov [C], edx

        push eax
        push resStr
        call [printf]

        push point
        call [printf]

        mov ebx, 0
        lp:

            mov eax, [C]
            mov ecx, [B]
            imul eax, 10

            mov edx, 0
            div ecx
            mov [C], edx

            push eax
            push emptyStr
            call [printf]

            add ebx, 1
            cmp ebx, 3
        jne lp

        jmp finish
    notDiv:



    finish:

    call [getch]

    push NULL
    call [ExitProcess]
```

```
section '.idata' import data readable
        library kernel, 'kernel32.dll',\
                msvcrt, 'msvcrt.dll'

        import kernel,\
               ExitProcess, 'ExitProcess'

        import msvcrt,\
               printf, 'printf',\
               scanf, 'scanf',\
               getch, '_getch'
```