

**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук

Департамент программной инженерии

**Микропроект №1
Программа определения количества чисел Ферма от 1 до беззнакового двойного
машинного слова.
ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

Исполнитель
Студент группы БПИ 199
_____/Мостачев А.О./
«__»_____2020 г.

Москва

2020

Содержание

Содержание	2
1. Постановка задачи	3
2. Применяемые расчетные методы и хранение данных	4
3. Входные и выходные данные	5
4. Используемые источники	6
5. Приложение 1	7
6. Приложение 2	10

1. Постановка задачи

Разработать программу определения количества чисел Ферма от 1 до беззнакового двойного машинного слова. Программа должна быть разработана на языке ассемблер.

2. Применяемые расчетные методы и хранение данных

Для возведения числа 2 в степень был использован побитовый сдвиг влево. Вычисление новых чисел Ферма происходило, пока вновь найденное число не превысит введенную верхнюю границу поиска или не станет меньше предыдущего числа. Второе условие контролировало переполнение беззнакового двойного машинного слова.

Введенное число сохраняется в переменной `urper`, для сохранения изменений количества подсчитанных чисел Ферма в ходе итерации используется регистр `ebx`, на каждой итерации подсчитанное число Ферма сохраняется в переменной `prev`.

3. Входные и выходные данные

На вход программой принимается число в интервале [1, 4294967295]. 0 на вход не принимается из-за формулировки поставленной задачи. При некорректном вводе числа оно будет преобразовано к числу из описанного интервала. При введении строки, не являющейся числом, она принимается за 0 и далее не обрабатывается.

Программа выводит одно число – количество чисел Ферма от 1 до введенного числа – с поясняющей строкой. Так как в задании не было строго обговорено, мной было решено включить введенное пользователем число в интервал поиска. Таким образом, установлено следующее соответствие между входными и выходными данными:

Введенное число входит в интервал	Полученное выходное значение
[1, 2]	0
[3, 4]	1
[5, 16]	2
[17, 256]	3
[257, 65536]	4
[65537, 4294967295]	5

Если ввод был некорректным, программа выводит соответствующее сообщение.

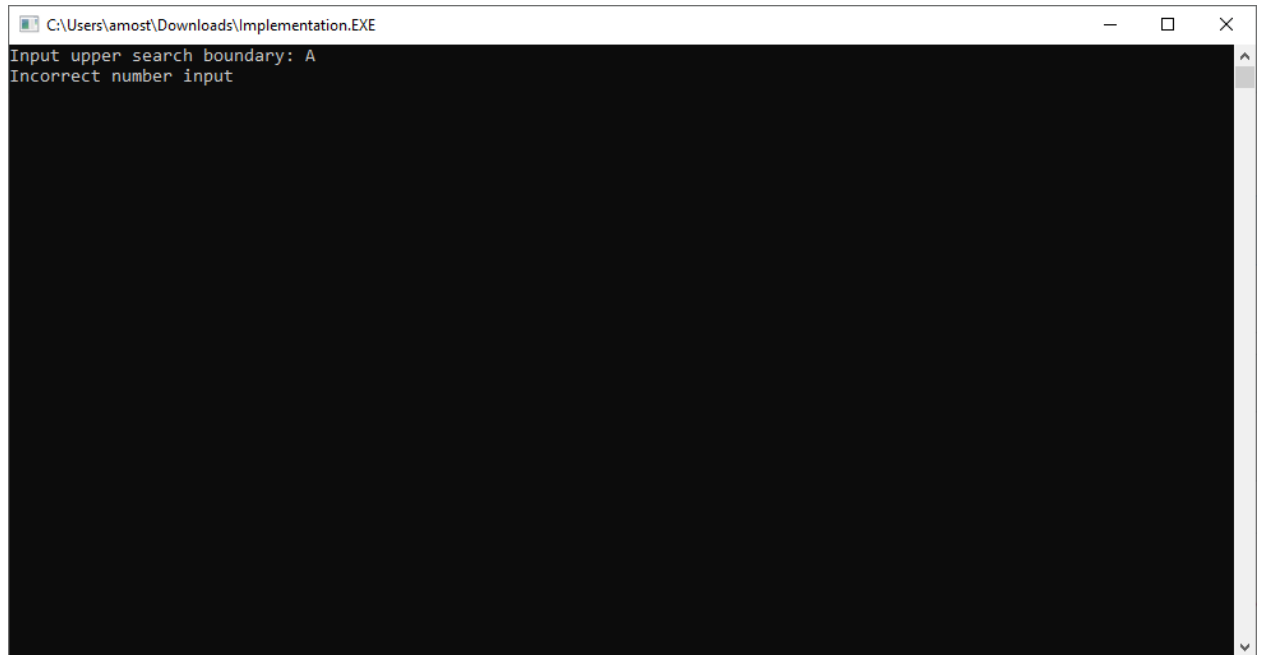
4. Используемые источники

- 1) Побитовый сдвиг в FASM
https://c9x.me/x86/html/file_module_x86_id_285.html
- 2) Числа Ферма
https://ru.wikipedia.org/wiki/%D0%A7%D0%B8%D1%81%D0%BB%D0%BE_%D0%A4%D0%B5%D1%80%D0%BC%D0%B0

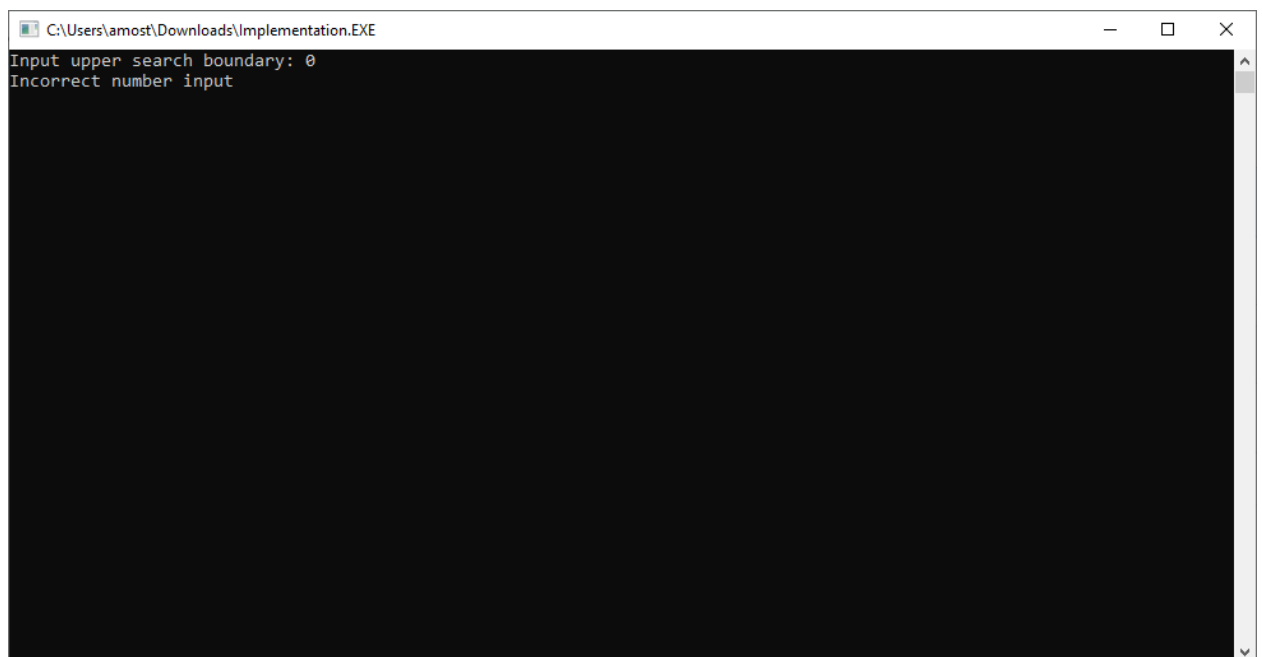
5. Приложение 1

Тестирование программы

1) Некорректный ввод

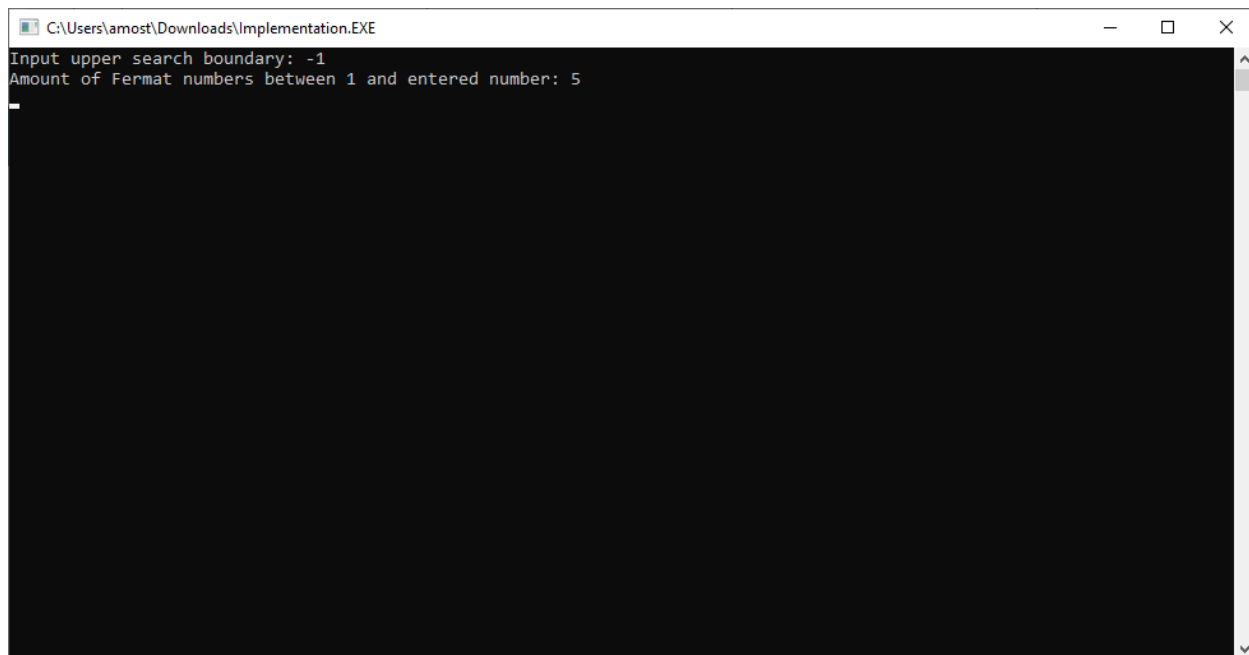


```
C:\Users\amost\Downloads\Implementation.EXE
Input upper search boundary: A
Incorrect number input
```



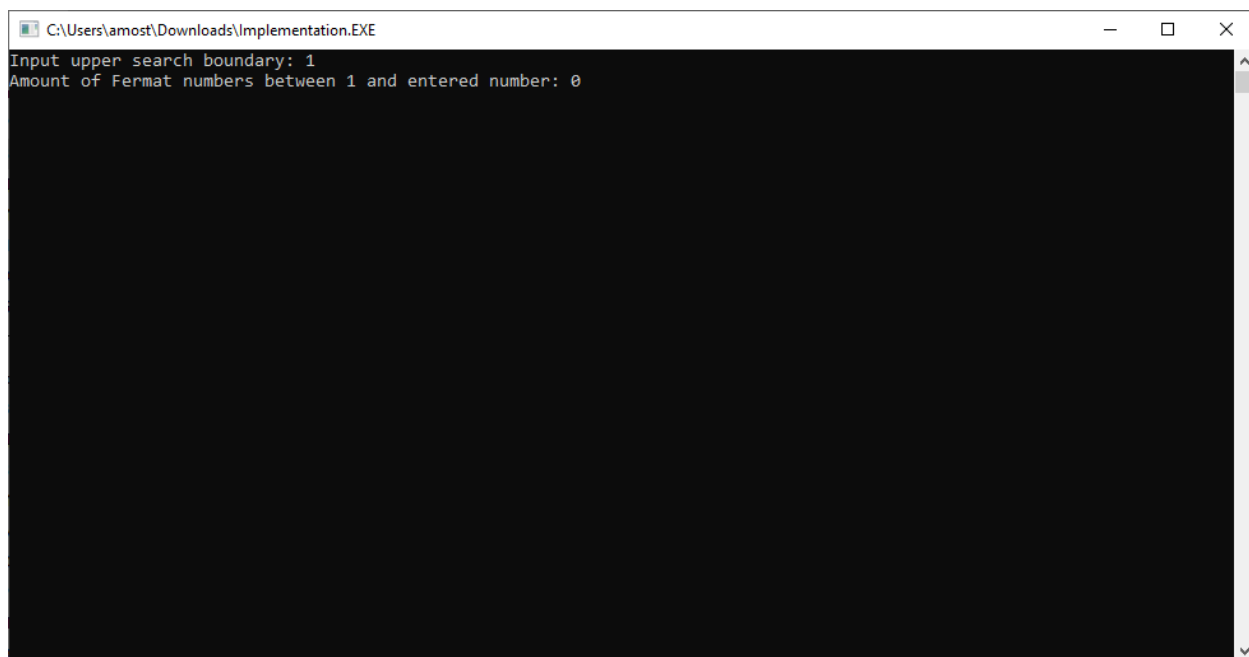
```
C:\Users\amost\Downloads\Implementation.EXE
Input upper search boundary: 0
Incorrect number input
```

2) Ввод числа не из запрошенного интервала



```
C:\Users\amost\Downloads\Implementation.EXE
Input upper search boundary: -1
Amount of Fermat numbers between 1 and entered number: 5
```

3) Ввод корректного значения



```
C:\Users\amost\Downloads\Implementation.EXE
Input upper search boundary: 1
Amount of Fermat numbers between 1 and entered number: 0
```



```
C:\Users\amost\Downloads\Implementation.EXE
Input upper search boundary: 5
Amount of Fermat numbers between 1 and entered number: 2
```

```
C:\Users\amost\Downloads\Implementation.EXE
Input upper search boundary: 4294967295
Amount of Fermat numbers between 1 and entered number: 5
```

6. Приложение 2

Текст программы

;Так как в задании не было замечено обратное,
;количество чисел Ферма считается до верхней границы включительно!

format PE console

entry start

include 'win32a.inc'

;-----

section '.data' data readable writable

strNum db 'Input upper search boundary: ', 0
strIncorSize db 'Incorrect number input', 0
strFermat db 'Amount of Fermat numbers between 1 and entered
number: %u', 10, 0
strScanInt db '%u', 0

upper dd 0

prev dd 0

;-----

section '.code' code readable executable

start:

call NumInput

call Fermat

finish:

call [getch]

push 0

call [ExitProcess]

;-----

;Считывает верхнюю границу поиска из консоли, сохраняя ее в переменную upper.
;При некорректном вводе вызывает функцию-обработчик failedNum.
;Затем записывает в регистр ebx -1 для подсчета количества чисел Ферма.
;При корректном вводе программа переходит к подсчету запрашиваемого значения.

NumInput:

```
    push strNum
    call [printf]
    add esp, 4

    push upper
    push strScanInt
    call [scanf]
    add esp, 8

    mov eax, [upper]
    cmp eax, 0
    je  failedNum

    mov ebx, -1

    ret
```

;Итерируется, пока новое найденное число Ферма не окажется больше введенного
;или новое число станет меньше предыдущего (так проверяется переполнение).
;Новое число Ферма на каждой итерации записывается в регистр edi.
;Для возведения в степень используется побитовый сдвиг влево.
;Регистр ecx используется для корректной работы сдвига.

Fermat:

```
    inc ebx
    mov ecx, ebx
    mov edi, 1
    mov esi, 1
```

```

    shl esi, cl
    mov ecx, esi

    shl edi, cl
    inc edi

    ;Если произошло переполнение, переходит к выводу.
    cmp edi, [prev]
    jb DisplayNum

    ;Последнее вычисленное число Ферма сохраняется в переменную temp.
    mov [prev], edi

    ;Итерация, пока не превзойдем введенное число.
    cmp [upper], edi
    jae Fermat

;Вывод подсчитанного количества чисел Ферма.
DisplayNum:
    push ebx
    push strFermat
    call [printf]
    add esp, 8
    ret

;-----
;Вывод сообщения о некорректном вводе.
failedNum:
    push strIncorSize
    call [printf]
    call [getch]
    push 0
    call [ExitProcess]

;-----

```

```

section '.idata' import data readable
    library kernel, 'kernel32.dll',\
        msvcrt, 'msvcrt.dll',\
        user32, 'USER32.DLL'

include 'api\user32.inc'
include 'api\kernel32.inc'
    import kernel,\
        ExitProcess, 'ExitProcess',\
        HeapCreate, 'HeapCreate',\
        HeapAlloc, 'HeapAlloc'
include 'api\kernel32.inc'
    import msvcrt,\
        printf, 'printf',\
        scanf, 'scanf',\
        getch, '_getch'

```