

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук
Департамент программной инженерии
Дисциплина: «Архитектура вычислительных систем»

**КОНСОЛЬНОЕ ПРИЛОЖЕНИЕ, ОПРЕДЕЛЯЮЩЕЕ
ПРОСТЫЕ ЧИСЛА ФИБОНАЧЧИ В ДИАПАЗОНЕ ОТ 1
ДО МАШИННОГО СЛОВА**

Пояснительная записка

Выполнил:
Петропавловский Андрей
студент гр. БПИ198.

31 октября 2020 г.

Москва
2020

Содержание

1. Текст задания	2
2. Применяемые расчетные методы	3
2.1. Теория решения задания.....	3
2.2. Дополнительный функционал программы	3
3. Тестирование программы.....	4
ПРИЛОЖЕНИЕ 1. Список литературы	5
ПРИЛОЖЕНИЕ 2. Код программы.....	6

1. Текст задания

Разработать программу, определяющую число простых чисел Фибоначчи в диапазоне от 1 до машинного слова.

2. Применяемые расчетные методы

2.1. Теория решения задания

В условии задачи нужно было найти простые числа Фибоначчи, так что я реализовал нахождение чисел последовательности способом динамического программирования, сохраняя последнее и предпоследнее найденное число в переменные, также реализовал проверку на простоту при каждом нахождении последующего числа в заданном диапазоне.

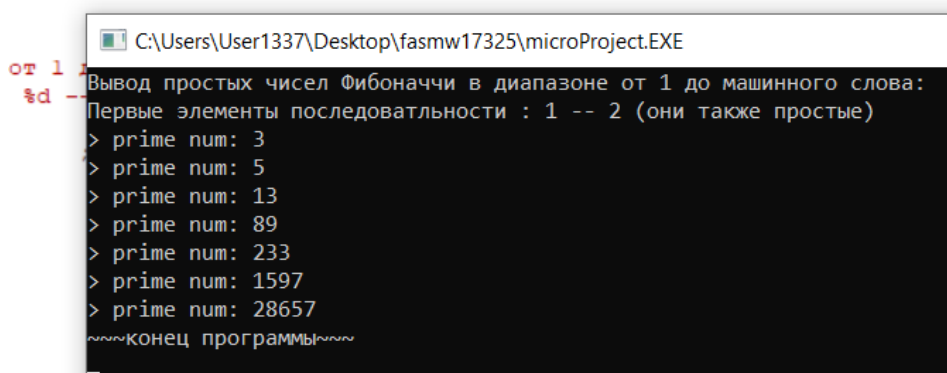
2.2. Дополнительный функционал программы

У этой задачи в принципе нету какого-то функционала, кроме подсчета чисел и вывода сообщений на экран пользователя, то есть никаких проверок входных данных осуществлено не было.

3. Тестирование программы

При запуске программы пользователю выводится сообщение о смысле программы и все числа, соответствующие заданию.

Последним выводится число 28567, так как следующее число последовательности (46368) непростое, а элементы дальше – больше машинного слова.



```
от 1
%d -
C:\Users\User1337\Desktop\fasmw17325\microProject.EXE
Вывод простых чисел Фибоначчи в диапазоне от 1 до машинного слова:
Первые элементы последовательности : 1 -- 2 (они также простые)
> prime num: 3
> prime num: 5
> prime num: 13
> prime num: 89
> prime num: 233
> prime num: 1597
> prime num: 28567
~~~конец программы~~~
```

1. Рисунок 1. Результат работы программы

ПРИЛОЖЕНИЕ 1**Список литературы**

1. Теория по нахождению числа Фибоначчи. [Электронный ресурс].
<https://habr.com/ru/post/261159> (дата обращения: 26.10.2020, свободный доступ).
2. Руководство fasm. [Электронный ресурс.]
<http://flatassembler.narod.ru/fasm.htm> (дата обращения: 29.10.2020, свободный доступ)

Код программы

```
;-----Петропавловский Андрей БПИ198-----  
;-----ВАРИАНТ 24-----
```

```
; Текст задания:
```

```
;Разработать программу, определяющую число простых чисел Фибоначчи  
;в диапазоне от 1 до машинного слова
```

```
format PE console
```

```
entry start
```

```
include 'win32a.inc'
```

```
section '.data' data readable writable
```

```
ru db 'Russian',0
```

```
i dd 0
```

```
f1 dd ?
```

```
f2 dd ?
```

```
d dd ?
```

```
temp dd ?
```

```
num dd ?
```

```
probel db ' ',13,10,0
```

```
info db 'Вывод простых чисел Фибоначчи в диапазоне от 1 до машинного  
слова: ',13,10,0 ;т.е. до 65535
```

```
showTwoNums db 'Первые элементы последовательности: %d — %d (они  
также простые) ',13,10,0
```

```
showNum db '> prime num: %d ',13,10,0
```

```
probell db '(%d) ',13,10,0
```

```
enddb '~~~конец программы~~~',13,10,0
```

```
section '.code' code readable executable
```

```
start:
```

```
cinvoke setlocale,0,ru
```

```
call searchFibo
```

```
push 0
```

```
call [getch]
```

```
call [ExitProcess]
```

```

proc searchFibo
    invoke printf, info
    add esp, 4
    mov [f1], 1
    mov [f2], 2
    mov ecx, [f1]
    mov edx, [f2]
    push [f2]           ;Вывод первого сообщения
    push [f1]
    push showTwoNums
    call [printf]
    add esp, 12
    mov ebx, 0
    mov eax, 0

    loopFiba:

        mov ecx, dword[f1]
        mov edx, dword[f2]
        xor ebx, ebx
        add ebx, ecx
        add ebx, edx ;Сумма предпоследнего и последнего числа ->
следующее число последовательности
        mov [num], ebx

        mov [d], 1
        forEach:
            xor ecx, ecx
            xor edx, edx
            inc [d]
            mov eax, dword[num]
            mov ebx, dword[d]
            div ebx ;ForeEach для проверки числа на простоту
            cmp edx, 0
            jne forEach
            cmp ebx, dword[num]
            je continuePrime
            cmp ebx, dword[num]
            jne continue

```


;Метка для перезаписи двух последних значений, если следующее число последовательности простое + вывод этого числа

continuePrime:

```

    mov ecx,dword[f1]
    mov edx, dword[f2]
    mov [temp],edx
    mov ecx,edx ;f1=f2
    mov edx,dword[num] ;f2 = num = (f1+f2)(Подсчитан в начале метки
loopFiba)
    mov [f1], ecx
    mov [f2], edx      ;Перезапись последнего и предпоследнего числа
    inc [i]

    push [num]
    push showNum
    call [printf]
    add esp, 8
    cmp [i], 24
    jne loopFiba
    cmp [i], 24
    je finish

```

;Метка для перезаписи двух последних значений, если следующее число последовательности непростое

continue:

```

    mov ecx,dword[f1] ;b=f1
    mov edx, dword[f2]
    mov [temp],edx

    mov ecx,edx ;f1=f2
    mov edx,dword[num] ;f2 = num = (f1+f2)( Подсчитан в начале метки
loopFiba)
    mov [f1], ecx
    mov [f2], edx      ; Перезапись последнего и предпоследнего числа
    inc [i]
    mov eax, [i]
    cmp [i], 24
    jne loopFiba      ;Использую кол-во чисел Фибоначии, т.к. 25ое
число последовательности > машинного слова т.е. 75025 > 65535

```

```

cmp [i], 24
je finish

```

```

finish:

```

```

    push endd
    call [printf]      ;Завершение программы
    add esp, 4

```

```

ret
endp

```

```

section '.idata' import data readable

```

```

library kernel,'kernel32.dll',\
    msvcrt, 'msvcrt.dll'

```

```

import kernel,\
    ExitProcess,'ExitProcess'      ;Импортируемые библиотеки

```

```

import msvcrt,\
    setlocale, 'setlocale',\
    printf, 'printf',\
    system,'system',\
    getch, '_getch'

```