

Федеральное государственное автономное образовательное учреждение высшего
образования

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

Факультет компьютерных наук

САМОСТОЯТЕЛЬНАЯ РАБОТА

Разработать программу определяющей максимальное значение параметра
числа линейной рекуррентной последовательности $t_n = t_{n-1} + t_{n-2} + t_{n-3} + t_{n-4}$ при $n \geq 4$
со стартовой последовательностью чисел $[0, 1, 0, 1]$, которое не выходит за
пределы беззнакового двойного машинного слова.

по направлению подготовки Архитектура вычислительных систем
образовательная программа «Программная инженерия»

Выполнил:

Ефимов Даниил

Александрович, студент

БПИ-195

Преподаватель:

Легалов Александр Иванович,

Доктор технических наук,

профессор

Москва 2020

Задание

Разработать программу определяющей максимальное значение параметра числа линейной рекуррентной последовательности $t_n = t_{n-1} + t_{n-2} + t_{n-3} + t_{n-4}$ при $n \geq 4$ со стартовой последовательностью чисел $[0, 1, 0, 1]$, которое не выходит за пределы беззнакового двойного машинного слова.

Составление программы

В данной программе сначала просится ввод n с помощью метода `scanf` и n не может быть меньше 4, это мы проверяем с помощью функции `cmp` и `js`, потому что если меньше, то мы завершаем программу, переходя к коду окончания программы `loopernd`. Далее мы инициализируем переменные `t_1`, `t_2`, `t_3`, `t_4` и `t_n` и после переходим к главному коду с помощью функции `call` в нашем случае `Main`.

В начале `Main` присваиваем регистрам нулевые значения, а после переходим в цикл `loopS`, в котором мы для начала добавляем в регистр все переменные `t_1`, `t_2`, `t_3`, `t_4`, а далее с помощью функции `mov` меняем местами и в `t_4` записываем уже новый элемент, а после инкрементируем переменную `i` и выводим `t_n`. Далее мы проверяем равны ли `i` и введенное n , и если они равны переходим к окончанию программы.

В окончании мы пользуемся импортированными функциями `getch` и `ExitProcess`.

Что касается вывода, то для вывода используется `'%u'`, потому что мы выводим беззнаковое двойное машинное слово.

А максимальный элемент в данном случае будет 37, так беззнаковое двойное машинное слово имеет пределы от 0 до $2^{32}-1$, то есть при начальных значениях 0, 1, 0, 1 максимальное число получается 2574579487 на 37 позиции.

Текст программы:

```
format PE console
entry start

include 'win32a.inc'

section '.data' data readable writable
    e1 db '%u', 10, 0
    strNumber db 'Enter n = ', 0
    strScanInt db '%d', 0
    strI db 'n = %d    t_n =', 0
    num dd 0
    t_1 dd 0d
    t_2 dd 1d
    t_3 dd 0d
    t_4 dd 1d
    real df 0d
    i dd 4 ;for loop
    t_n dd 0
    slashn db 10, 0, 0
    tt df 0
    numerating dd ' ', 0
    lastel db 'n = 37    t_n = %u', 10, 0
    num1 dd 0
```

section '.code' code readable executable

```
start:
    push strNumber
    call [printf]
    push num
    push strScanInt
    call [scanf] ;Enter n

    ;check for more than 4 n
    mov eax,4
    cmp eax, [num]
    jg loopend

    mov dword [t_1],0 ;Setting the initial value for the
first element
    mov dword [t_2],1 ;Setting the initial value for the
second element
    mov dword [t_3],0 ;Setting the initial value for the
third element
    mov dword [t_4],1 ;Setting the initial value for the
fourth element
    mov dword [t_n],0 ;Setting the initial value for the n
element

    call Main
```

Main:

```
;initialization edx and ecx
mov edx, 0
mov ecx, 0
loopS:

    mov eax, [t_1] ;add a variable t_1
    Jo loopend

    add eax, [t_2] ;add a variable t_2
    Jo loopend

    add eax, [t_3] ;add a variable t_3
    Jo loopend

    add eax, [t_4] ;add a variable t_4
    Jo loopend

    mov dword [t_n], eax

    ;write in t_1
    mov ecx, [t_1]
    mov ebx, [t_2]
    mov [t_1], ebx

    ;write in t_2
    mov ecx, [t_2]
    mov ebx, [t_3]
    mov [t_2], ebx

    ;write in t_3
    mov ecx, [t_3]
    mov ebx, [t_4]
    mov [t_3], ebx

    ;write in t_4
    mov ecx, [t_4]
    mov ebx, [t_n]
```

```

        mov [t_4], ebx

        ;increment i loop
        inc [i]

        ;output
        cinvoke printf, strI, [i]
        cinvoke printf, numerating
        cinvoke printf, el, [t_n]

        mov eax, [num] ;write to register n
        cmp eax, [i] ;compare register and n
        je loopend ;if equal then finish

        jmp loopS ;new iteration

loopend:

        mov eax, [num] ;write to register n
        cmp eax, [i] ;compare register and n
        jg writeanother ;if bigger
        call [getch]
        push 0
        call [ExitProcess]

writeanother:

        mov eax, 0
        add eax, [t_1]
        add eax, [t_2]
        add eax, [t_3]
        add eax, [t_4]
        cinvoke printf, lastel, eax
        call [getch]
        push 0
        call [ExitProcess]

section '.idata' import data readable

        library kernel, 'kernel32.dll',\
                msvcrt, 'msvcrt.dll',\
                user32, 'user32.dll'

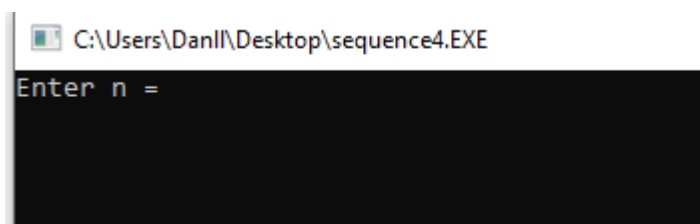
        import msvcrt,\
                printf, 'printf',\
                scanf, 'scanf',\
                getch, '_getch'

        import kernel,\
                GetProcessHeap, 'GetProcessHeap',\
                ExitProcess, 'ExitProcess'

```

Тестирование программы

Программа имеет параметр n, который мы вводим в самом начале:



Далее при разных значениях мы получаем разные ответы:

C:\Users\DanII\Desktop\sequence4.EXE

Enter n = 12

n = 5	t_n = 2
n = 6	t_n = 4
n = 7	t_n = 7
n = 8	t_n = 14
n = 9	t_n = 27
n = 10	t_n = 52
n = 11	t_n = 100
n = 12	t_n = 193

C:\Users\DanII\Desktop\sequence4.EXE


Enter n = 24

n = 5	t_n = 2
n = 6	t_n = 4
n = 7	t_n = 7
n = 8	t_n = 14
n = 9	t_n = 27
n = 10	t_n = 52
n = 11	t_n = 100
n = 12	t_n = 193
n = 13	t_n = 372
n = 14	t_n = 717
n = 15	t_n = 1382
n = 16	t_n = 2664
n = 17	t_n = 5135
n = 18	t_n = 9898
n = 19	t_n = 19079
n = 20	t_n = 36776
n = 21	t_n = 70888
n = 22	t_n = 136641
n = 23	t_n = 263384
n = 24	t_n = 507689

```
C:\Users\Danil\Desktop\sequence4.LAL
Enter n = 100
n = 5    t_n = 2
n = 6    t_n = 4
n = 7    t_n = 7
n = 8    t_n = 14
n = 9    t_n = 27
n = 10   t_n = 52
n = 11   t_n = 100
n = 12   t_n = 193
n = 13   t_n = 372
n = 14   t_n = 717
n = 15   t_n = 1382
n = 16   t_n = 2664
n = 17   t_n = 5135
n = 18   t_n = 9898
n = 19   t_n = 19079
n = 20   t_n = 36776
n = 21   t_n = 70888
n = 22   t_n = 136641
n = 23   t_n = 263384
n = 24   t_n = 507689
n = 25   t_n = 978602
n = 26   t_n = 1886316
n = 27   t_n = 3635991
n = 28   t_n = 7008598
n = 29   t_n = 13509507
n = 30   t_n = 26040412
n = 31   t_n = 50194508
n = 32   t_n = 96753025
n = 33   t_n = 186497452
n = 34   t_n = 359485397
n = 35   t_n = 692930382
n = 36   t_n = 1335666256
n = 37   t_n = 2574579487
```

А при неверном вводе программа ничего не выводит и завершает выполнение:

```
C:\Users\Danil\Desktop\sequence4.EXE
Enter n = 1
```

 C:\Users\DanII\Desktop\sequence4.EXE

Enter n = 3