

ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
"ВЫСШАЯ ШКОЛА ЭКОНОМИКИ"

Факультет компьютерных наук
Департамент программной инженерии

Микропроект_1
по дисциплине
Архитектура вычислительных систем

Пояснительная записка

Проверяющий:
Профессор ДПИ ФКН, д.т.н.
Легалов А.И.

Исполнитель:
студент группы БПИ198
Юдина Е. Д.
«27» октября 2020 г.

Текст задания:

29	Юдина Екатерина Дмитриевна	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 $[0, 0, 0, 1]$, которое не выходит за пределы беззнакового машинного слова
----	----------------------------	--

Программа выполнена с помощью знаний, полученных на лекциях и на семинарах по дисциплине ABC. Также использовались данные с сайта : <http://www.softcraft.ru/>

Комментарии к работе:

Программа должна выполнять элементарную рекурсию до тех пор, пока значение функции меньше или равно UInt32.MaxValue (максимальное значение беззнакового машинного слова). По окончании выполнения вывести n, соответствующее последнему вычисленному t_n. Для удобства пользователя выводятся все промежуточные значения n и соответственно значения функции.

Текст Программы (основные секции):

Ссылка на GitHub: <https://github.com/KateJud/MicroproectABS>

Секция данных

```
section '.data' data readable writable

    ru db 'Russian',0 ; Для русской локали

    string db 'n: %u fn=: %u',13,10,0 ;Строка для вывода промежуточных значения для пользователя
    ;u -целое без знака
    res db 'Ответ: n=%u максимальное n, при котором не будет переполнения.',13,10,0 ;
    t1 dd 0 ;t_n-1-ый элемент
    t2 dd 0 ;t_n-2
    t3 dd 0 ;t_n-3
    t4 dd 1 ;t_n-4
    tn dd ? ;n-ый элемент
    ans dd 3 ;n - начинаем с 4( но +=1 делаем перед выводом, при первом проходе должно быть 4, поэтому устанавливаем знач 3

    NULL = 0
```

Секция кода

```
section '.code' code readable executable

start:
    cinvoke setlocale,0,ru ;Устанавливаем русскую локаль
    mov eax,0
    mov [tn],0           ;Обнуляем текущий элемент
    mov eax, [t1]         ;Записываем в регистр для последующих вычислений
    add [tn], eax         ;Складываем текущий элемент с первым (tn=t1)
    jc output             ;Переходим к метке(output) если было переполнение
                           ;проверяем после каждого сложения
    mov eax, [t2]         ;Записываем в регистр для последующих вычислений
    add [tn], eax         ;складываем текущий элемент со вторым (tn=t1+t2)
    jc output             ;проверяем на переполнение
    mov eax, [t3]         ;Записываем в регистр для последующих вычислений
    add [tn], eax         ;складываем текущий элемент со вторым (tn=t1+t2+t3)
    jc output             ;проверяем на переполнение
    mov eax, [t4]         ;Записываем в регистр для последующих вычислений
    add [tn],eax          ;складываем текущий элемент со вторым (tn=t1+t2+t3+t4)
    jc output             ;проверяем на переполнение
    inc[ans]              ;Если переполнения не было, то увеличиваем n
                           ;Сдвигаем элементы: t1<-t2, t2<-t3, t3<-t4; t4<-tn
    mov eax,[t2]
    mov [t1],eax
    mov eax, [t3]
    mov [t2], eax
    mov eax, [t4]
    mov [t3], eax
    mov eax, [tn]
    mov [t4], eax
    push [t4]             ;Выводим f_n
    push [ans]            ;и n
    push string           ;на экран после вычисления суммы на каждом этапе рекурсии
    call [printf]

    jmp start             ;Повторяем заново

output:                  ;выводим номер последнего n на экран
    push [ans]
    push res
    call [printf]

    call [getch]          ;"Замораживаем" экран
    push NULL
    call [ExitProcess]
```

Результат работы программы:

```
n: 4 fn=: 1
n: 5 fn=: 2
n: 6 fn=: 4
n: 7 fn=: 8
n: 8 fn=: 15
n: 9 fn=: 29
n: 10 fn=: 56
n: 11 fn=: 108
n: 12 fn=: 208
n: 13 fn=: 401
n: 14 fn=: 773
n: 15 fn=: 1490
n: 16 fn=: 2872
n: 17 fn=: 5536
n: 18 fn=: 10671
n: 19 fn=: 20569
n: 20 fn=: 39648
n: 21 fn=: 76424
n: 22 fn=: 147312
n: 23 fn=: 283953
n: 24 fn=: 547337
n: 25 fn=: 1055026
n: 26 fn=: 2033628
n: 27 fn=: 3919944
n: 28 fn=: 7555935
n: 29 fn=: 14564533
n: 30 fn=: 28074040
n: 31 fn=: 54114452
n: 32 fn=: 104308960
n: 33 fn=: 201061985
n: 34 fn=: 387559437
n: 35 fn=: 747044834
n: 36 fn=: 1439975216
n: 37 fn=: 2775641472
Ответ: n=37 максимальное n, при котором не будет переполнения.
```

P.S. Была выполнена проверка и при $n = 38$:

$f_n = 5'350'220'959 > 4'294'967'295 = \text{UInt32.MaxValue}$

Следовательно программа работает вено и выводит максимальное n , при котором не происходит переполнение.