

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

Факультет компьютерных наук  
Департамент программной инженерии

**Микропроект №1  
Пояснительная записка**

**Листов 7**

Гогарев Игорь,  
БПИ192, подгруппа 2

## Оглавление

Текст задания .....	2
Расчетные методы.....	2
Результат работы .....	3
Текст программы .....	4

### Текст задания

Разработать программу, определяющей максимальное значение параметра числа линейной рекуррентной последовательности  $t_n = t_{n-1} + t_{n-2} + t_{n-3}$  при  $n \geq 3$  ("числа трибоначчи") со стартовой тройкой чисел  $[0, 1, 1]$ , которое не выходит за пределы машинного слова

### Расчетные методы

При разработке программы использовались знания о последовательности Фибоначчи. Вычисления проводились по аналогии с последовательностью Фибоначчи, только использовалась сумма последних трех элементов для вычисления каждого следующего элемента.

Для текущей рассматриваемой «тройки» использовались три переменные –  $a$ ,  $b$ ,  $c$ .

Переход к следующей тройке осуществлялся следующим образом: в отдельную переменную  $sum$  записывалась сумма  $a+b+c$ . Далее в переменную  $a$  записывалось значение  $b$ , в переменную  $b$  записывалось значение  $c$ , в переменную  $c$  записывалось значение  $sum$ .

Элементы последовательности выводятся по одному в строку в цикле, условие выхода из цикла выполняется тогда, когда очередное значение  $sum$  стало меньше значению переменной  $b$  (аналогично наступлению переполнения).

## Результат работы

C:\Users\1\Documents\fasmw17325\test1.EXE

```
[1] -> 0
[2] -> 1
[3] -> 1
[5] -> 2
[6] -> 4
[7] -> 7
[8] -> 13
[9] -> 24
[10] -> 44
[11] -> 81
[12] -> 149
[13] -> 274
[14] -> 504
[15] -> 927
[16] -> 1705
[17] -> 3136
[18] -> 5768
[19] -> 10609
[20] -> 19513
[21] -> 35890
[22] -> 66012
[23] -> 121415
[24] -> 223317
[25] -> 410744
[26] -> 755476
[27] -> 1389537
[28] -> 2555757
[29] -> 4700770
[30] -> 8646064
[31] -> 15902591
[32] -> 29249425
[33] -> 53798080
[34] -> 98950096
[35] -> 181997601
[36] -> 334745777
[37] -> 615693474
[38] -> 1132436852
[39] -> 2082876103
The results are:
[f_1] = 615693474
[f_2] = 1132436852
[f_3] = 2082876103
Max number before being overflow: 2082876103 = 334745777 + 615693474 + 1132436852.
First number after overflow is -463960867.
```

## Текст программы

```

;Гогарев Игорь БПИ192 Вариант 6
format PE console
entry start

include 'include/win32a.inc'

section '.data' data readable writable
    resultsStr      db 'The results are:',10, 0
    strMax          db 'Max number before being overflow: %d = %d + %d +
%d. ',10, 0
    firstOverflow   db 'First number after overflow is %d.', 0
    strScanInt      db ' [%d] -> %d ',10, 0 ;для вывода элементов
последовательности трибоначчи с их порядковым номером
    f1Elem          db '[f_1] = %d', 10, 0 ;для вывода первого из трех
последних элементов до переполнения
    f2Elem          db '[f_2] = %d', 10, 0 ;для вывода второго из трех
последних элементов до переполнения
    f3Elem          db '[f_3] = %d', 10, 0 ;для вывода третьего из трех
последних элементов до переполнения
    tmpStack        dd ? ;будет содержать указатель на стек при входе в
подпрограмму
    a               dd 0 ;первый элемент рассматриваемой тройки
чисел
    b               dd 0 ;второй элемент рассматриваемой тройки чисел
    c               dd 0 ;третий элемент рассматриваемой тройки чисел
    sum             dd 0 ;содержит сумму a, b, c
    reserve         dd 0 ;будет хранить в подпрограмме вывода результатов
наименьшее слагаемое в максимальной сумме
    counter         dd 0 ;счетчик для вычисления порядкового элемента в
последовательности трибоначчи
    NULL = 0
section '.code' code readable executable

start:

    call initialization
    call loop_output
    call printResults

finish:

    call [getch]
    push NULL
    call [ExitProcess]

;-----ПОДПРОГРАММЫ-(SUBPROGRAMS)-----
initialization:
    ;ниже проходит вывод в консоль первых трех элементов
последовательности трибоначчи - 0, 1, 1

```

```

        ;и их порядковый номер
        mov [tmpStack], esp
        push 0
        push 1
        push strScanInt
        call[printf]
        push 1
        push 2
        push strScanInt
        call[printf]
        push 1
        push 3
        push strScanInt
        call[printf]
        ;ниже происходит инициализация первой тройки - 0, 1, 1, а
        порядковый номер становится равным 4
        mov eax, 0
        mov [a], eax
        mov eax, 1
        mov [b], eax
        mov [c], eax
        mov [counter], 4
        mov esp, [tmpStack]
        ret

```

loop\_output:

```

        mov [tmpStack], esp
        ;здесь в цикле выводятся числа последовательности трибоначчи вместе с
        соответствующим порядковым номером
        ;условия выхода из цикла - когда очередная сумма(sum)станет меньше второго
        элемента тройки(b)
        lp:
        mov eax,[a]
        add eax, [b]
        add eax, [c]
        mov [sum], eax
        mov eax, [sum]
        cmp eax, [b]
        jl printResults
        mov eax, [b]
        mov [a], eax
        mov eax, [c]
        mov [b], eax
        mov eax, [sum]
        mov [c], eax
        push [c]
        mov eax, [counter]
        inc eax
        push eax
        mov [counter], eax
        push strScanInt
        call [printf]

        jmp lp
        mov esp, [tmpStack]

```

ret

printResults:

;здесь происходит вывод результатов вычислений

mov [tmpStack], esp

push resultsStr

call[printf]

mov eax, [c]

sub eax, [b]

sub eax, [a]

mov [reserve], eax ;теперь в reserve записана разность между последней положительной суммой и двумя наибольшими слагаемыми

;таким образом, reserve - это наименьший элемент максимальной суммы

;ниже происходит вывод последних трех элементов последовательности трибоначчи до переполнения

mov eax,[a]

push eax

push f1Elem

call [printf]

mov eax,[b]

push eax

push f2Elem

call [printf]

mov eax,[c]

push eax

push f3Elem

call [printf]

;ниже происходит вывод наибольшей суммы и того, как она получилась

push [b]

push [a]

push [reserve]

push [c]

push strMax

call[printf]

;ниже происходит вычисления следующей суммы (она уже будет переполнена)  
и ее вывод

mov eax, [c]

add eax, [a]

add eax, [b]

push eax

push firstOverflow

call [printf]

mov esp, [tmpStack]

jmp finish

ret

;-----

section '.idata' import data readable

library kernel, 'kernel32.dll',\

msvcrt, 'msvcrt.dll'

import kernel,\

ExitProcess, 'ExitProcess'

import msvcrt,\

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