

**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»
Факультет компьютерных наук
Департамент программной инженерии**

**Микропроект № 1
по дисциплине
"Архитектура вычислительных систем"**

Пояснительная записка

Исполнитель:
Студент группы БПИ191
/ Самаренко А. В./
«31» октября 2020 г.

Москва 2020

АННОТАЦИЯ

В данном программном документе приведена пояснительная записка к микропроекту № 1. по дисциплине "Архитектура вычислительных систем".

Данная пояснительная записка содержит в себе следующие разделы:

- В разделе «Введение» указано наименование программы и документы, на основании которых ведется разработка.
- В разделе «Расчетные методы» указана теоретическая составляющая программы, а также некоторые необходимые для полного понимания алгоритма процессы и решения (описание вывода, алгоритма вычисления элемента последовательности и т.д.)
- В разделе "Дополнительный функционал" указаны косметические и функциональные возможности программы сверх требуемых согласно техническому заданию проекта
- В разделе "Описание входных данных" представлено описание входных данных разработанной программы
- Приложение содержит скриншоты исходного кода программы (в текстовом виде код представлен в репозитории)

Содержание

1. ВВЕДЕНИЕ	6
1.1. Наименование программы.....	6
1.2. Документы, на основании которых ведётся разработка	6
1.3. Описание	6
2. РАСЧЕТНЫЕ МЕТОДЫ.....	7
2.1. Подсчёт члена линейной рекуррентной последовательности	7
2.1.1. Определение линейной рекуррентной последовательности	7
2.1.2. Вычисление следующего члена рекуррентной последовательности.....	7
2.2. Описание переменных.....	7
2.3. Схема работы алгоритма	8
2.4. Выход за граничное значение	8
2.5. Вывод результата.....	8
3. ДОПОЛНИТЕЛЬНЫЙ ФУНКЦИОНАЛ	9
3.1. Вывод каждого члена рекуррентной последовательности.....	9
3.2. Вывод минимального значения параметра числа рекуррентной последовательности.....	9
4. ОПИСАНИЕ ВХОДНЫХ ДАННЫХ	10
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ	11
ПРИЛОЖЕНИЕ	12

1. ВВЕДЕНИЕ

1.1. Наименование программы

- Наименование программы – «Микропроект № 1».

1.2. Документы, на основании которых ведётся разработка

- Программа выполнена в рамках итогового задания по дисциплине "Архитектура вычислительных систем", в соответствии с учебным планом подготовки бакалавров по направлению 09.03.04 «Программная инженерия», 2 курс 1 модуль.
- Основанием для разработки является письмо профессора факультета компьютерных наук Легалова Александра Ивановича от 07.10.20

1.3. Описание

- Программа вычисляет максимальное значение параметра числа линейной рекуррентной последовательности $f(n) = f(n + 2) - f(n + 1)$, при $n \leq -2$ ("числа Фибоначчи") со стартовой последовательностью $[0, 1]$ в отрицательной области значений, не выходящее за пределы целого со знаком $= 10^9$

2. РАСЧЕТНЫЕ МЕТОДЫ

2.1. Подсчёт члена линейной рекуррентной последовательности

2.1.1. Определение линейной рекуррентной последовательности

Рекуррентная последовательность - всякая числовая последовательность x_0, x_1, \dots , задаваемая линейным рекуррентным соотношением: $x_n = a_1 * x_{n-1} + \dots + a_d * x_{n-d}$ для всех $n \geq d$ с заданными начальными членами $x_0 \dots x_{d-1}$, где d - фиксированное натуральное число $a_0 \dots a_d$ - заданные числовые коэффициенты, $a_d \neq 0$.

2.1.2. Вычисление следующего члена рекуррентной последовательности

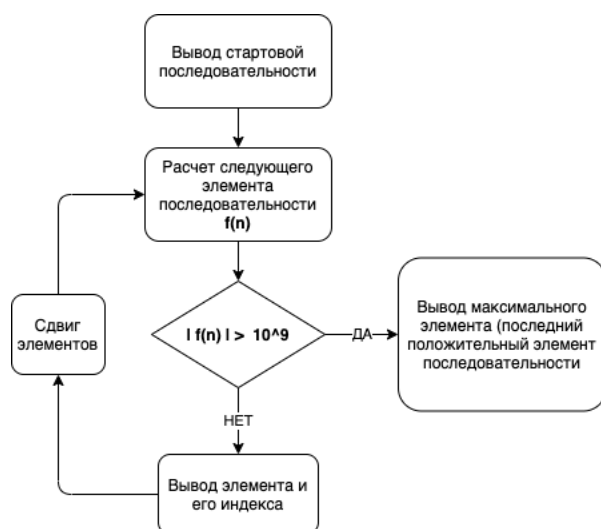
В техническом задании данного программного продукта приведено конкретное рекуррентное соотношение: $f(n) = f(n+2) - f(n+1)$, при $n \leq -2$ согласно которому вычисляется любой член данной последовательности, стартовая последовательность равна $[0,1]$.

$f(0) = 1, \quad f(-1) = 1, \quad f(-2) = f(0) - f(-1) = 0 - 1 = -1$ и так далее

2.2. Описание переменных

Переменная	Тип	Значение	Смысл
strScanInt	db	'%d'	вывод члена последовательности
strVecElemI	db	'[F(%d)]: -----> '	индекс члена последовательности
strScanInt	db	"	перевод строки (разделитель)
minElem	db	'Min Element of the sequence: = %d'	вывод минимального члена
maxElem index	db	'Max Element of the sequence: = %d'	вывод максимального члена
index	dd	?	хранение порядка последовательности
maxAbs	dd	1000000000	абсолютная граница значений
n_n0	dd	0	предыдущий член
n_n1	dd	1	текущий член

2.3. Схема работы алгоритма



(Рис. 1 Алгоритм работы программы)

2.4. Выход за граничное значение

При вычислении последующего члена рекуррентной последовательности необходимо сравнивать полученное значение с граничным (по условию 10^9), чтобы избежать выхода за пределы допустимых значений. Заданная рекуррентная последовательность является знакопеременной, поэтому сравнивать значения необходимо по абсолютной величине (по модулю).

2.5. Вывод результата

При выполнении программы каждый член рекуррентной последовательности выводится в отдельную строчку в формате "[F(n)]" = value(F(n)). При переполнении (выходе за допустимую область значений) выводится результат - максимальное значение параметра числа, а также минимальное значение (дополнительный функционал). Ответ и сама последовательность разделяются пустой строкой.

3. ДОПОЛНИТЕЛЬНЫЙ ФУНКЦИОНАЛ

3.1. Вывод каждого члена рекуррентной последовательности

Программа выводит на экран не только максимальное значение параметра числа, но и каждое значение рекуррентной последовательности, начиная со стартовой последовательности.

3.2. Вывод минимального значения параметра числа рекуррентной последовательности

Программа рассчитывает не только максимальное, но и минимальное значение параметра числа данной рекуррентной последовательности.

4. ОПИСАНИЕ ВХОДНЫХ ДАННЫХ

Исходя из поставленной задачи, входные данные для выполнения программы не требуются.

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

- 1) Assembler.com.ua [Электронный ресурс] // Assembler.com.ua: [сайт]. [2020] URL: assembler.com.ua/, режим доступа: свободный, дата обращения 25.10.2020
- 2) FLAT ASSEMBLER 1.64 - Мануал [Электронный ресурс] // <http://flatassembler.narod.ru/>: [сайт]. [2020] URL: <http://flatassembler.narod.ru/fasm.htm>, режим доступа: свободный, дата обращения 25.10.2020
- 3) SoftCraft [Электронный ресурс] // SoftCtaft: [сайт]. [2020] URL: <http://softcraft.ru/edu/comparch/>, режим доступа: свободный, дата обращения 25.10.2020
- 4) StackOverflow [Электронный ресурс] // StackOverflow: [сайт]. [2020] URL: <https://ru.stackoverflow.com/>, режим доступа: свободный, дата обращения 25.10.2020
- 5) Byte++ [Электронный ресурс] // Byte++: [канал]. [2020] URL: <https://www.youtube.com/channel/UCG7GW-X1cczyzLswoYTTnjQ>, режим доступа: свободный, дата обращения 25.10.2020

```

format PE console
entry start

include 'win32a.inc'

;-----
section '.data' data readable writable

    strVecElemI db '[F(%d)]: -----> ', 0 ;строка для представления n-ого члена рекуррентной последовательности
    strScanInt db '%d', 13, 10, 0 ;строка для вывода значения n-ого члена рекуррентной последовательности
    nextStr db ', 13, 10, 0 ;пустая строка для разделения данных
    minElem db 'Min Element of the sequence: = %d', 13, 10, 0 ;строка для представления пользователю информации о минимальном элементе рекуррентной
    maxElem db 'Max Element of the sequence: = %d', 13, 10, 0 ;строка для представления пользователю информации о максимальном элементе рекуррентной

    index dd ? ;итерируемая переменная для подсчета номера члена последовательности (n)
    ;программа также может работать для любого числа в отрезке [1;2^31-1]
    maxAbs dd 1000000000 ;граница, заданная по условию 10^9
    n_n0 dd 0 ;f(n+2)
    n_n1 dd 1 ;f(n+1)

;-----Основная часть программы-----
section '.code' code readable executable
start:
    call SpecialFib
finish:
    call [getch]

    push 0
    call [ExitProcess]

SpecialFib:

    xor eax, eax ;очищаем регистры для использования в нашей функции
    xor ebx, ebx ;данный регистр предназначен для хранения f(n+2)
    xor ecx, ecx ;данный регистр предназначен для хранения f(n+1)
    xor edx, edx ;данный регистр предназначен для хранения индекса (n)

    ;инициализируем первые два элемента последовательности для вычисления последующих чле
    ;f(0) = 0
    ;f(-1) = 1
    ;f(n) = f(n+2) - f(n+1)

;f(n) = f(n+2) - f(n+1)

InitSequence:

    mov [index], eax ;инициализируем индекс нулевым значением

    ;производим печать индекса в специальной форме

    push [index]
    push strVecElemI
    call [printf]
    add esp, 8

    ;производим печать значения f(0) = 0

    push [n_n0]
    push strScanInt
    call [printf]
    add esp, 8

    ;сдвигаем индекс на -1

    mov eax, [index]
    dec eax
    mov [index], eax

    ;производим печать значения f(-1) = 1

    push [index]
    push strVecElemI
    call [printf]
    add esp, 8

    push [n_n1]
    push strScanInt
    call [printf]
    add esp, 8

;-----Основной цикл программы-----
printLoop:

    ;мы точно знаем, что в данный ряд - знакопеременный - каждый элемент под четным индексом
    ;поэтому мы каждый раз проверяем на переполнение

    mov ebx, [n_n1]

    ;уменьшаем индекс

    mov eax, [index]

```

```

mov eax, [index]                ;уменьшаем индекс
dec eax
mov [index], eax

mov eax, [n_n0]                 ;производим вычисление элемента
sub eax, ebx

mov edx, eax                    ;если полученное значение больше нуля - приведение к модулю не требуется

cmp eax, 0
jge comparison                  ;записываем модуль в регистр edx

neg eax

comparison:                     ;сравниваем с граничным значением значение текущего элемента последовательности

cmp eax, [maxAbs]
jge endPrint

mov [n_n1], edx                 ;происходит сдвиг элементов в последовательности
;используем edx до вызова функций (строго!) ;в регистре edx хранится элемент f(n+1)

mov [n_n0], ebx                 ;в регистре ebx хранится элемент f(n+2)

push [index]                    ;печать индекса
push strVecElemI
call [printf]
add esp, 8

push [n_n1]                     ;печать самого значения элемента последовательности
push strScanInt
call [printf]
add esp, 8

inc ecx                         ;увеличиваем наш каунтер на 1, чтобы наш цикл не завершился преждевременно

loop printLoop                  ;используем loop согласно рекомендации

;-----Вывод программы-----
endPrint:

push nextStr
call [printf]
add esp, 4                       ;разделитель между выводом последовательности и ответом

cmp [n_n0], 1
jge firstMaxThenMin             ;выбор типа вывода в зависимости от знака элемента последовательности

firstMinThenMax:
push [n_n0]
push minElem
call [printf]
add esp, 8

push [n_n1]
push maxElem
call [printf]
add esp, 8

ret

firstMaxThenMin:
push [n_n1]
push minElem
call [printf]
add esp, 8

push [n_n0]
push maxElem
call [printf]
add esp, 8

ret

;-----HeapApi-----

section '.idata' import data readable
library kernel, 'kernel32.dll',\

;-----HeapApi-----

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'

```