

**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
Власюк 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. Выход за граничное значение	7
2.3. Вывод результата	Ошибка! Закладка не определена.
3. ДОПОЛНИТЕЛЬНЫЙ ФУНКЦИОНАЛ	9
3.1. Вывод каждого члена рекуррентной последовательности	10
3.2. Вывод минимального значения параметра числа рекуррентной последовательности.....	Ошибка! Закладка не определена.
4. ОПИСАНИЕ ВХОДНЫХ ДАННЫХ.....	Ошибка! Закладка не определена.
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 задания по дисциплине "Архитектура вычислительных систем", в соответствии с учебным планом подготовки бакалавров по направлению 09.03.04 «Программная инженерия», 2 курс, 1 семестр.
- Основанием для разработки является письмо профессора факультета компьютерных наук Легалова Александра Ивановича от 07.10.20

1.3. Описание программы

- Программа определяет количество чисел Коллатца, не превышающих величины беззнакового двойного двойного машинного слова.

.

2. РАСЧЕТНЫЕ МЕТОДЫ

2.1. Подсчёт кол-ва чисел Каллена, не превышающих величины беззнакового двойного машинного слова

Для выполнения этой задачи, в программе организовывается цикл while. На каждом шаге цикла вычисляется очередное число Каллена и сравнивается с беззнаковым двойным машинным словом. Если данное число Каллена его превышает – то цикл завершается. Иначе – счётчик увеличивается и цикл продолжается.

2.2. Вычисление чисел Каллена

2.2.1. Определение

Числа Каллена – натуральные числа вида $n * 2^n + 1$.

2.2.2. Вычисление следующего члена рекуррентной последовательности

На каждом шаге i алгоритма, решающего задачу программы, мы вычисляем значение соответствующей степени двойки (2^i), домножив степень, вычисленную на предыдущем шаге алгоритма, на 2. Затем умножаем его на i и прибавляем к результату 1 – получается число Каллена под номером i .

2.3. Выход числа Каллена за границы двойного машинного слова

При вычислении очередного числа Каллена, может оказаться, что оно занимает больше места, чем двойное машинное слово. Тогда программа также завершается, так как это число Каллена больше любого двойного машинного слова.

2.4. Описание переменных

Переменная	Тип	Значение	Смысл
strInputNum	db	'Input an unsigned dword: '	Указание пользователю ввести число
strUintFormat	db	'%u'	Формат беззнакового целого числа
strIncorrectInput	db	Сообщение пользователю при некорректном вводе	'The input is not a number!'
strPressAnyKey	db	'Press any key to exit...'	Указание пользователю нажать любую клавишу
two	dd	2	Двойка
powerOfTwo	dd	1	Текущая степень двойки (вычисляется в цикле)
inputNum	dd	?	Введенное пользователем число
counter	dd	0	Счетчик

2.5. Описание подпрограмм

Подпрограмма	Цель
CheckInput	Проверяет корректность пользовательского ввода. В случае некорректного ввода программа завершается.
FindCullenNumbersQuantityLoop	С помощью инструкции loop вычисляет кол-во чисел Каллена, не превышающих значение беззнакового двойного машинного слова.

3. ОПИСАНИЕ ВХОДНЫХ ДАННЫХ

На вход в программу поступает беззнаковое целое двойное машинное слово, которое пользователь вводит в консоль.

4. ДОПОЛНИТЕЛЬНЫЙ ФУНКЦИОНАЛ

4.1. Проверка корректности ввода

В программе организована проверка корректности ввода – если пользователь ввёл не целое число, то программа выводит сообщение и завершается.

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

- 1) Assembler.com.ua [Электронный ресурс] // Assembler.com.ua: [сайт]. [2020] URL: assembler.com.ua/, режим доступа: свободный, дата обращения 20.10.2020
- 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 20.10.2020
- 3) SoftCraft [Электронный ресурс] // SoftCtaft: [сайт]. [2020] URL: <http://softcraft.ru/edu/comparch/>, режим доступа: свободный, дата обращения 20.10.2020
- 4) StackOverflow [Электронный ресурс] // StackOverflow: [сайт]. [2020] URL: <https://ru.stackoverflow.com/>, режим доступа: свободный, дата обращения 26.10.2020
- 5) Byte++ [Электронный ресурс] // Byte++: [канал]. [2020] URL: <https://www.youtube.com/channel/UCG7GW-X1ccyzLswoYTTnjQ>, режим доступа: свободный, дата обращения 19.10.2020

ПРИЛОЖЕНИЕ

1) Секция с данными

```
flat assembler 1.73.25
File Edit Search Run Options Help
format PE console
entry start

include 'win32a.inc'

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

    strInputNum    db 'Input an unsigned dword: ', 0
    strUIntFormat  db '%u', 0
    strIncorrectInput db 'The input is not a number!', 10, 0
    strPressAnyKey db 10, 'Press any key to exit...', 0

    two            dd 2
    powerOfTwo     dd ? ; Power of two.
    inputNum       dd ? ; Number that the user inputs.
    counter        dd ? ; Number of Cullen numbers that are not greater than inputNum.

    null = 0
;-----
```

2) Секция импорта

```
;-----
section '.code' code readable executable

start:
    ; Input
    push strInputNum
    call [printf]

    push inputNum
    push strUIntFormat
    call [scanf]

    ; Check for correct input
    call CheckInput

    ; Calculate result
    call FindCullenNumbersQuantityLoop

    ; Output
    push [counter]
    push strUIntFormat
    call [printf]

finish:
    ; Wait for user to press a key
    push strPressAnyKey
    call [printf]

    call [getch]

    push null
    call [ExitProcess]
;-----
```

3) Вызов методов и подпрограмм

```
;-----  
section '.code' code readable executable  
  
start:  
    ; Input  
    push strInputNum  
    call [printf]  
  
    push inputNum  
    push strUIntFormat  
    call [scanf]  
  
    ; Check for correct input  
    call CheckInput  
  
    ; Calculate result  
    call FindCullenNumbersQuantityLoop  
  
    ; Output  
    push [counter]  
    push strUIntFormat  
    call [printf]  
  
finish:  
    ; Wait for user to press a key  
    push strPressAnyKey  
    call [printf]  
  
    call [getch]  
  
    push null  
    call [ExitProcess]  
;-----
```

4) Подпрограмма для проверки корректности ввода

```
;-----  
CheckInput:  
    cmp eax, 1          ; Check that user input is one number  
    jne incorrectInput  
  
    ;Else:  
    ret  
  
incorrectInput:  
    push strIncorrectInput    ; Print a message and finish the program  
    call [printf]  
  
    jmp finish  
;-----
```

- 5) Подпрограмма, реализующая подсчет чисел Каллена, не превышающих величины беззнакового двойного машинного слова

```
FindCullenNumbersQuantityLoop:
    mov [powerOfTwo], 1
    mov [counter], null

    mov ecx, 1

continueLoop:

    ; Calculating C_n (Number of cullen)
    mov eax, [counter]
    mul [powerOfTwo] ; mul writes result to eax and edx

    ; If edx is not null then result of mul is greater than a dword
    cmp edx, null
    jne endOfLoop

    ; Else continue
    mov ebx, eax
    inc ebx

    ; Check if current Cullen number is not greater than inputNum
    cmp ebx, [inputNum]
    ja endOfLoop

    ; Else increase counter and continue
    add [counter], 1

    mov eax, [two]
    mul [powerOfTwo]
    mov [powerOfTwo], eax ; Calculate power of two

    ; So that the loop won't finish early
    inc ecx

endOfLoop:
    loop continueLoop
    ret
```