

**Правительство Российской Федерации**

**Федеральное государственное автономное образовательное**

**учреждение высшего образования «Национальный**

**исследовательский университет «Высшая школа экономики»**

**Факультет компьютерных наук**

**Департамент программной инженерии**

**Отчет к домашнему заданию По дисциплине**

**«Архитектура вычислительных систем»**

**Работу выполнил:**

**Студент группы БПИ-196 Мосолков Е.Н.**

**Москва 2020**

## Задача

16	Мосолков Евгений Николаевич	Разработать программу определения количества чисел Каллена, не превышающего величины беззнакового двойного машинного слова
----	-----------------------------	--

## Решение

Считаем количество чисел Каллена (числа вида  $n * (2^n) + 1$ ) с точностью до двойного машинного слова (UINT –  $2^{32}$  бит - 4294967296)

Для подсчета числа каллена храним счетчик  $i$  в памяти (по формуле выше  $i$  – это  $n$ ). Возводим это число в степень командой `shl` и храним степень двойки в памяти (за это отвечает переменная `power`), затем умножаем командой `mul` степень двойки на  $i$ , затем командой `add` добавляем 1 и получаем число Каллена (текущее).

Команда `mul ebx` записывает в регистр `eax` произведение регистров `eax` и `ebx`

Команда `shl eax, 1` – делаем побитовый сдвиг влево на 1, т.е. возводим на каждой итерации 2 в степень большую чем на предыдущей итерации

Команда `add eax, 1` – добавляет единицу к регистру `eax`

Далее на каждой итерации мы сохраняем предыдущее число Каллена, для того, чтобы проверить переполнение регистра, если предыдущее число Каллена меньше текущего, то мы выводим количество чисел в консоль. Если же условие не выполнено то мы переходим к следующей итерации.

Отсчет ведется с  $i = 0$ , значит при выводе следует добавить 1, т.к. 1 – тоже считается числом Каллена ( $0 * 2^0 + 1$ ).

Изначально предыдущее число Каллена равно 1, т.к. запоминаем мы это число только в конце.

Инкремент переменной  $i$ , происходит в начале подпрограммы, значит текущее число Каллена – всегда больше 1.

Подпрограмма `CountCullenNumbers` – считает количество чисел каллена до  $2^{32}$  и записывает ответ в переменную  $i$

Подпрограмма `Output` – выводит сообщение о выводе и количество чисел Каллена не превышающих двойного машинного слова.

## Тестирование

Посчитаем количество чисел Каллена не превышающих  $2^{32}$  на высокоуровневом языке Python. Получаем число 28.

```
In [2]: count = 0
        i = 0
        cond = True
        while cond == True:
            cullenNum = i * 2**i + 1
            if cullenNum < 2**32:
                count+=1
                i+=1
            else:
                cond = False
        count|

Out[2]: 28
```

Рисунок 1.

```
Count of CullenNumbers: 28_
```

Рисунок 2.

При запуске программы cullen\_numbers.exe (скомпилированный файл cullen\_numbers.asm) получаем число 28, что является корректным ответом.

## Текст программы

```
format PE console
```

```
entry start
```

```
include 'win32a.inc'
```

```
;-----
```

```
section '.data' data readable writable
```

```
    strCullenNumberCount    db 'Count of CullenNumbers: %d', 0
```

```
    cullenNumber            dd 0
```

```
    prevCullen              dd 1
```

```
    i                       dd 0
```

```
    power                   dd 1
```

```
;-----
```

```
section '.code' code readable executable
```

```
start:
```

```
    jmp Count
```

```
finish:
```

```
    call [getch]
```

```
    push 0
```

```
    call [ExitProcess]
```

```
;-----
```

```
Output:
```

```
    ; because we count cullen numbers from i = 0, we adds 1 extra number
```

```
    mov eax, [i]
```

```
    inc eax
```

```
    mov [i], eax
```

```
    ; now we do output in console
```

```
    push [i]
```

```
    push strCullenNumberCount
```

```
    call [printf]
```

```
    add esp, 8
```

```
    jmp finish
```

```

;-----
Count:
    ; i++
    mov eax, [i]
    inc eax
    mov [i], eax
    ; increse calculate 2^i
    mov ebx, [power]
    shl ebx, 1
    mov [power], ebx
    ; calculate current cullen number: i*(2^i) + 1
    mul ebx
    add eax, 1
    ; save last cullen number to check if number is greater than double dword
    cmp eax, [prevCullen]
    jl Output
    mov [prevCullen], eax
    jmp Count

;-----third act - including 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'

```

## **Список используемых источников**

1. Википедия (2020) «Число Каллена»  
[https://ru.wikipedia.org/wiki/%D0%A7%D0%B8%D1%81%D0%BB%D0%BE\\_%D0%A4%D0%B5%D1%80%D0%BC%D0%B0](https://ru.wikipedia.org/wiki/%D0%A7%D0%B8%D1%81%D0%BB%D0%BE_%D0%A4%D0%B5%D1%80%D0%BC%D0%B0)
2. SoftCraft «Программирование на языке ассемблера. Микропроект. Требования к оформлению. 2020-2021 уч.г.» - <http://softcraft.ru/edu/comparch/tasks/mp01/>
3. Лекции и семинары курса 3 семестра программной инженерии «Архитектуры вычислительных систем»