

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ  
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук. Департамент программной инженерии.  
Дисциплина: «Архитектура вычислительных систем»

**МИКРОПРОЕКТ №1**

Пояснительная записка

**Выполнила:**  
Кириченко Виктория  
*студент гр. БПИ198.*

**Москва**  
2020

## Содержание

1. Текст задания .....	2
2. Применяемые расчетные методы .....	3
2.1. Теория решения задания.....	3
2.2. Дополнительный функционал программы .....	3
3. Тестирование программы .....	4
3.1. Корректные значения .....	4
3.2. Некорректные значения .....	4
ПРИЛОЖЕНИЕ 1. Список литературы .....	5
ПРИЛОЖЕНИЕ 2. Код программы.....	6

## **1. Текст задания**

«Разработать программу определения параметра максимального числа Вудала, не превышающего длины беззнакового машинного слова».

## **2. Применяемые расчетные методы**

### **2.1. Теория решения задания**

Необходимо подсчитать числа Вудала до тех пор, пока значение не превысит  $2^{32}-1$ , где число Вудала рассчитывается по следующей формуле:  $\{W_{\{n\}}=n * 2^{\{n\}}-1\}$ . И вывести последнее  $n$ , при котором  $W_{\{n\}}$  удовлетворяет условию.

### **2.2. Дополнительный функционал программы**

Вывод не только подходящего параметра, но и всех чисел, подсчитанных во время исполнения программы. Плюс вывод самого максимального числа из заданного диапазона.

### 3. Тестирование программы

#### 3.1. Корректные значения

Программа не подразумевает входные данные, поэтому был приведен вывод всех подсчитанных значений с указанием номера итерации, для отображения корректности работы программы.

На рис.1 видно, что 28 число Вудала вышло за границы  $2^{32}-1$ , т. к. оно стало меньше предыдущего. Следовательно берем только 27 параметр и само число.

```
Woodall numbers less 2^32-1
[1] = 1
[2] = 7
[3] = 23
[4] = 63
[5] = 159
[6] = 383
[7] = 895
[8] = 2047
[9] = 4607
[10] = 10239
[11] = 22527
[12] = 49151
[13] = 106495
[14] = 229375
[15] = 491519
[16] = 1048575
[17] = 2228223
[18] = 4718591
[19] = 9961471
[20] = 20971519
[21] = 44040191
[22] = 92274687
[23] = 192937983
[24] = 402653183
[25] = 838860799
[26] = 1744830463
[27] = 3623878655
[28] = 3221225471
Max Woodall number = [3623878655]
Parametr of max Woodall number n = [27]
```

1. Рисунок 1. Вывод всех чисел, включая максимальное

#### 3.2. Некорректные значения

Программа работает без входных значений.

### Список литературы

1. Типы данных в ассемблере. [Электронный ресурс] // URL: <https://prog-cpp.ru/asm-datatypes/> (дата обращения: 23.10.2020)
2. Число Вудала. [Электронный ресурс] // URL: [https://ru.wikipedia.org/wiki/%D0%A7%D0%B8%D1%81%D0%BB%D0%BE\\_%D0%92%D1%83%D0%B4%D0%B0%D0%BB%D0%B0](https://ru.wikipedia.org/wiki/%D0%A7%D0%B8%D1%81%D0%BB%D0%BE_%D0%92%D1%83%D0%B4%D0%B0%D0%BB%D0%B0) (дата обращения: 23.10.2020)
3. Unsigned integers in assembly. [Электронный ресурс] // URL: <https://stackoverflow.com/questions/42795616/unsigned-integers-in-assembly> (дата обращения: 26.10.2020)
4. Основы языка Assembler. [Электронный ресурс] // URL: <http://window.edu.ru/catalog/pdf2txt/303/78303/59184> (дата обращения: 26.10.2020)
5. Программирование на языке ассемблера. [Электронный ресурс] // URL: <http://natalia.appmat.ru/c&c++/assembler.html> (дата обращения: 28.10.2020)

**Код программы**

```

2. format PE console
3. entry start
4.
5. include 'win32a.inc'
6.
7. ;-----
8. section '.data' data readable writable
9.
10.     strInfo  db 'Woodall numbers less 2^32-1',10,0
11.     strVecElemI db '[%u] = ',0 ; number of index in array
12.     strScanInt db '%u',10,0 ;number of array vec[i](Woodall number)
13.     strVecElemOut db '[%u] = %u',10,0
14.     strWoodallNum db 'Max Woodall number = [%u]',10,0
15.     strWoodallPar db 'Parametr of max Woodall number n = [%u]',0
16.
17.     vec_size  dd 1
18.     sum        dd 0
19.     i          dd 1 ; iterator
20.     tmp         dd ? ;current number (vec[i])
21.     tmp2        dd 1 ;exp of 2
22.     tmp3        dd 1 ;vec[1] or vec[i-1]
23.     tmpStack    dd ?
24.     vec         rd 100 ;array of Woodall numbers
25.
26.;-----
27.section '.code' code readable executable

```

28.start:

29.; 1) generate array

30.     call GenerateArray

31.

32.finish:

33.     call [getch]

34.

35.     push 0

36.     call [ExitProcess]

37.

38.;-----

39.GenerateArray:

40.     push strInfo ; info about what numbers presented

41.     call [printf]

42.     add esp, 4

43.

44.getArray:

45.     mov ecx, 1             ; iterator from 1

46.     mov ebx, vec         ; array of Woodall numbers

47.     ; mov edx, 1         ; ebx = &vec

48.     mov eax, 1           ;exponentiation of 2

49.     ; mov esi, 0

50.

51.ArrayLoop:

52.     mov [tmp], ebx ; move value of array to tmp

53.     mov [i], ecx ; move iterator to i

54.

55.     push ecx



```

56.    push strVecElemI ; ordinal number, print it
57.    call [printf]
58.    add esp, 8
59.    mov eax, [tmp2] ; taking a number from tmp2 exp of 2 previous number
60.    imul eax, 2 ;increasing degree
61.    mov [tmp2], eax
62.    mov ebx, 1 ; vec[i] = 1(for right multiplication)
63.    imul ebx,[i] ; formula
64.    imul ebx,[tmp2] ; formula
65.    dec ebx ; formula , where vec[i] =  $i*2^{i-1}$ 
66.
67.    push ebx ;print calculated number
68.    push strScanInt
69.    call [printf]
70.    add esp, 8
71.
72.    cmp [tmp3], ebx ;if previous number of array < current number of array
73.    jnb PlusOne ;i++
74.    cmp [tmp3], ebx
75.    je PlusOne
76.
77.    jmp endGenerateArray ;if not, turn to the end
78.
79.
80.PlusOne:
81.    mov [tmp3], ebx ; remembering vec[i] in tmp3
82.    mov ecx, [i] ; jump to nex val of array
83.    inc ecx

```

```

84.    mov ebx, [tmp]
85.    add ebx, 4
86.    jmp ArrayLoop    ;continue
87.
88.endGenerateArray:
89.    push [tmp3]        ;latest appropriate number in array
90.    push strWoodallNum ; print it
91.    call [printf]
92.    add esp,8
93.    mov ecx,[i]        ;latest appropriate iterator always i-1
94.    dec ecx
95.    push ecx
96.    push strWoodallPar
97.    call [printf]
98.    add esp,8
99.    ret
100.
101.    ;-----third act - including HeapApi-----
    -----
102.
103.    section '.idata' import data readable
104.        library kernel, 'kernel32.dll',\
105.            msvcrt, 'msvcrt.dll',\
106.            user32,'USER32.DLL'
107.
108.    include 'api\user32.inc'
109.    include 'api\kernel32.inc'
110.    import kernel,\

```

```
111.      ExitProcess, 'ExitProcess',\  
112.      HeapCreate, 'HeapCreate',\  
113.      HeapAlloc, 'HeapAlloc'  
114.  include 'api\kernel32.inc'  
115.      import msvcrt,\  
116.      printf, 'printf',\  
117.      scanf, 'scanf',\  
      getch, '_getch'
```