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

Факультет компьютерных наук  
Департамент программной инженерии

**КОНСОЛЬНОЕ ПРИЛОЖЕНИЕ ДЛЯ ОПРЕДЕЛЕНИЯ ЧИСЛА ЭЛЕМЕНТОВ ПОСЛЕДОВАТЕЛЬНОСТИ ПАДОВАНА ОТ 1 ДО БЕЗЗНАКОВОГО МАШИННОГО СЛОВА**  
**Пояснительная записка**

Исполнитель  
Студент группы БПИ199  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Галанов А. С./  
«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2020 г.

**Москва 2020**

**Оглавление**

[Текст задания 3](#_Toc55161561)

[Применяемые расчетные методы 4](#_Toc55161562)

[П.1 4](#_Toc55161563)

[П.2 4](#_Toc55161564)

[Тестовый пример работы приложения 6](#_Toc55161565)

[Список использованной литературы 10](#_Toc55161566)

Текст задания

Разработать программу, определяющую число элементов последовательности Падована от 1 до беззнакового машинного слова

Применяемые расчетные методы

Программа делиться на два взаимосвязанных этапа:

1. Подсчет количества элементов последовательности для 32 разрядных систем
2. Подсчет количества элементов последовательности для 64 разрядных систем

П.1

В п.1 использовался макрос cinvoke, который через указатель, помеченный первым аргументом, вызывает нужную функцию из библиотеки msvcrt: printf. Данная функция предназначена для вывода соответствующей информации в консоль. В начале программы уведомляем пользователя о предназначении программы, а также выводим первые три элемента последовательности, поскольку они являются основой для подсчета следующих элементов и их считать не надо. Пользуясь определением последовательности Падована из Вики [1] считаем каждый следующий элемент. Для этого используем бесконечный цикл loop. Поскольку в конце мы кладем в регистр ecx число 2, а предикат выполнения цикла это “ecx-1>0 ?”, поэтому цикл никогда не закончится. Завершение цикла происходит с помощью инструкции jc, которая срабатывает если флаг CF=1. Если после выполнения инструкции add (сложение двух чисел) происходит переполнение, то флаг СF становится равный единице и происходит выход из цикла. Данное переполнение говорит о том, что следующие число последовательности является больше величины регистра, то есть больше чем 2^32, и данное число нам не надо учитывать при подсчете суммы всех элементов, так как оно больше машинного слова для процессоров с соответствующей архитектурой, к примеру, для 32-битных процессоров архитектуры x86. После подсчета количества элементов происходит вывод информации для пользователя и переход ко п.2

П.2

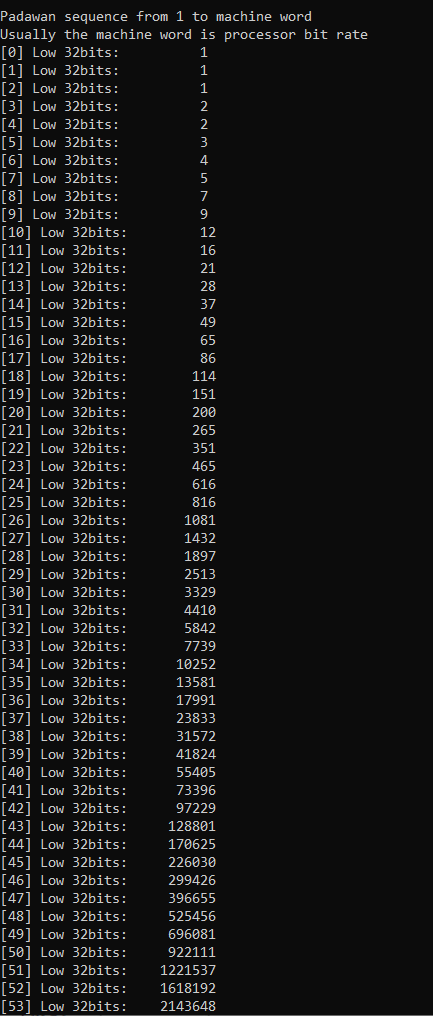
п.2 аналогичен п.1, но предназначен для подсчета количества элементов для машинного слова для 64-битных процессоров архитектуры х64. Смысл схож с п.1, присутствует бесконечный цикл(loop), а также сложение двух чисел для получения следующего числа последовательности, но в данном случае из за того, что числа наши теперь больше 2^32, а регистры хранят в себе только 32 бит информации, мы разбиваем каждое число на 32 старших бита и на 32 младших бита и по отдельности их складываем. Сдвиг в числе на старшие биты осуществляется с помощью прибавления числа 4 к указателю на ячейку памяти нашего числа (4, потому что 4 байт это 32 бит). После сложения младших битов, наш флаг СF может стать единицей, и значит мы должны прибывать 1 к страшим битам, данную функцию выполняет инструкция adc. После сложения младших битов, очищаем флаг CF, c помощью инструкции СLC (CLear Cf), и складываем старшие 32 бита. Если при сложении старших битов информации происходит переполнение, то флаг CF обратно ставится в положение равный единице и тогда с помощью инструкции jc происходит выход из цикла. В конце программы выводиться соответствующие сообщение о количестве чисел последовательности <=2^64 – машинное слово на 64 разрядных процессорах.

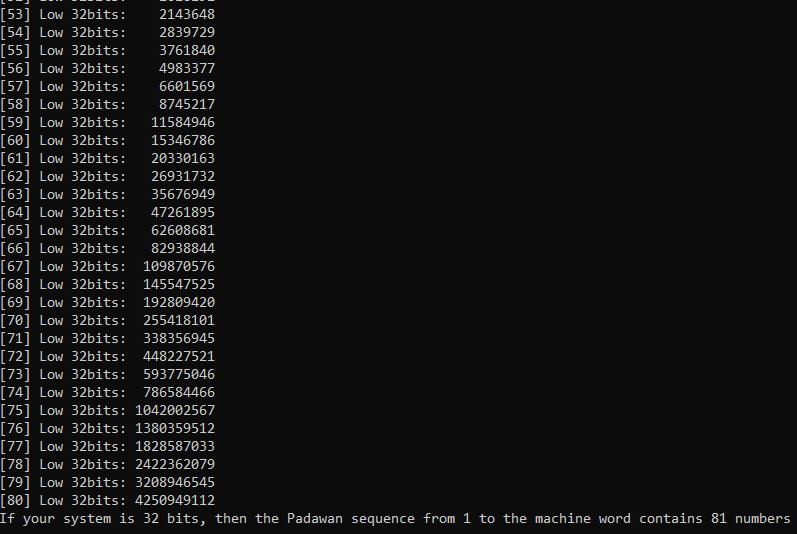
Вся информация по работе с инструкциями, метками и процедурами бралась из документации по FASM и assembler [3],[4]

Информация о машинных словах бралась из обучающей статьи [5]

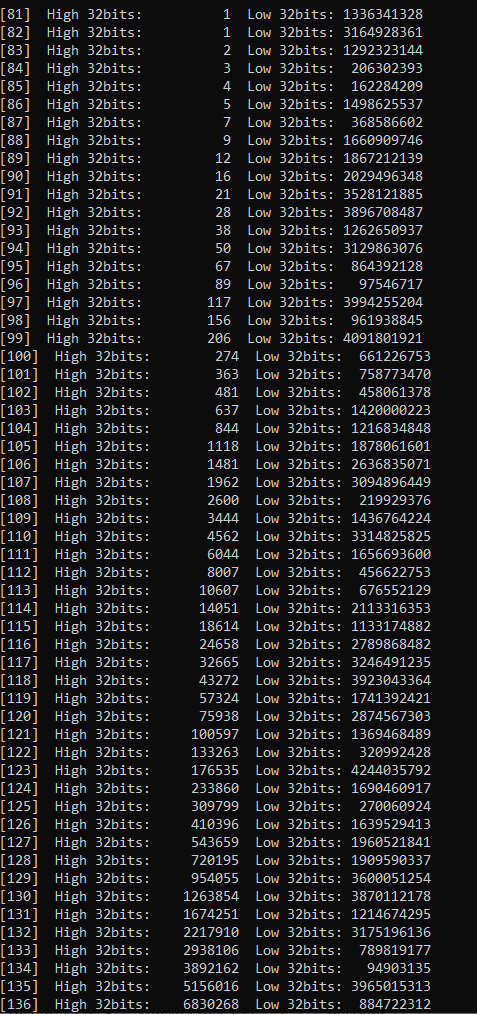
Тестовый пример работы приложения

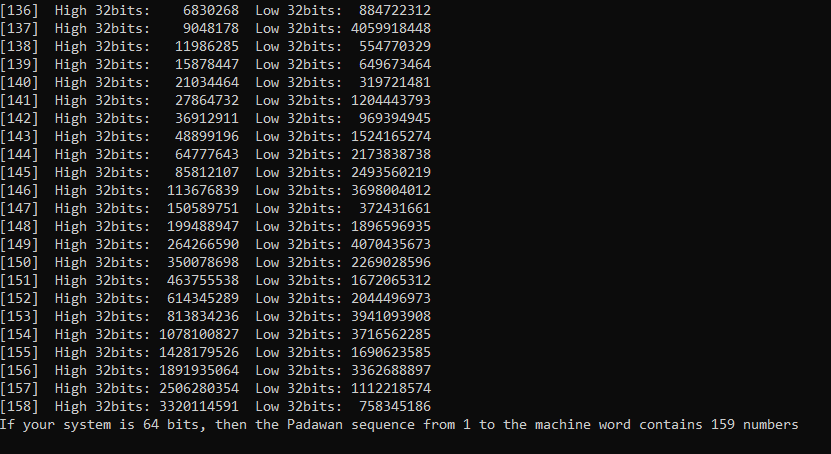
1. Вывод информации для 32 разрядных процессоров





1. Вывод информации для 64 разрядных процессоров





Пояснительная записка написана на основании инструкции [2].

Список использованной литературы

1. Обучающая статья про последовательность Падована [Электронный ресурс] //URL: [https://ru.wikipedia.org/wiki/Последовательность\_Падована](https://ru.wikipedia.org/wiki/Последовательность_Падована%20) (Дата обращения: 28.10.2020, режим доступа: свободный)
2. Инструкция по составлению пояснительной записки [Электронный ресурс]. //URL: [http://softcraft.ru/edu/comparch/tasks/mp01/](http://softcraft.ru/edu/comparch/tasks/mp01/%20) (Дата обращения: 28.10.2020, режим доступа: свободный)
3. Документация по ассемблеру [Электронный ресурс]. //URL: [https://flatassembler.net/docs.php?article=win32](https://flatassembler.net/docs.php?article=win32%20) (Дата обращения: 28.10.2020, режим доступа: свободный)
4. Основное руководство FASM [Электронный ресурс]. //URL: [http://flatassembler.narod.ru/fasm.htm](http://flatassembler.narod.ru/fasm.htm%20) (Дата обращения: 28.10.2020, режим доступа: свободный)
5. Обучающая статья про машинные слова [Электронный ресурс] //URL: [https://ru.wikipedia.org/wiki/Машинное\_слово#Размер\_машинного\_слова\_на\_различных\_архитектурах](https://ru.wikipedia.org/wiki/Машинное_слово%23Размер_машинного_слова_на_различных_архитектурах%20) (Дата обращения: 28.10.2020, режим доступа: свободный)