

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

Факультет компьютерных наук
Образовательная программа бакалавриата «Программная инженерия»

**ПРОГРАММА ДЛЯ ОПРЕДЕЛЕНИЯ МАКСИМАЛЬНОГО ЗНАЧЕНИЯ ПАРАМЕТРА
ФУНКЦИИ ФАКТОРИАЛА, ПРИ КОТОРОМ ЗНАЧЕНИЕ ФУНКЦИИ ФАКТОРИАЛА НЕ
ВЫХОДИТ ЗА ПРЕДЕЛЫ ПОЛОЖИТЕЛЬНОГО ЦЕЛОГО РАЗМЕРОМ В ДВОЙНОЕ
МАШИННОЕ СЛОВО**

Пояснительная записка

Исполнитель
студент группы БПИ196
_____ / Татаринов Н.А. /
«31» октября 2020 г.

Москва 2020

Пояснительная записка

Листов 15

Москва 2020

АННОТАЦИЯ

В данном программном документе приведена пояснительная записка к программе «Программа для определения максимального значения параметра функции факториала, при котором значение функции факториала не выходит за пределы положительного целого размером в двойное машинное слово».

В разделе «Введение» указано наименование программы и краткая характеристика области её применения.

В разделе «Технические характеристики» содержатся следующие подразделы:

- постановка задачи на разработку программы;
- описание алгоритма и функционирования программы с обоснованием выбора схемы алгоритма решения задачи и возможные взаимодействия программы с другими программами;
- описание и обоснование выбора метода организации входных и выходных данных.

В разделе «Код программы» исходный текст программы на языке Ассемблер с подробными комментариями необходимыми для понимания программы сторонним человеком.

Содержание

1. ВВЕДЕНИЕ	4
1.1. Наименование программы	4
1.2. Краткая характеристика области применения	4
2. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ	5
2.1. Постановка задачи на разработку программы	5
2.2. Описание алгоритма и функционирования программы	5
2.3. Описание и обоснование выбора метода организации входных и выходных данных	5
3. КОД ПРОГРАММЫ	6
4. ТЕСТИРОВАНИЕ	12
ПРИЛОЖЕНИЕ 1 ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ ПЕРЕМЕННЫХ	14
ПРИЛОЖЕНИЕ 2 ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ ПРОЦЕДУР	15

1. ВВЕДЕНИЕ

1.1. Наименование программы

Наименование программы – «Программа для определения максимального значения параметра функции факториала, при котором значение функции факториала не выходит за пределы положительного целого размером в двойное машинное слово».

1.2. Краткая характеристика области применения

Текст задания: «Разработать программу, которая определяет максимальное значение параметра функции факториала, при котором значение функции не выходит за пределы положительного целого размером в двойное машинное слово».

Программа выполняется в рамках курса «Архитектура вычислительных систем» в соответствии с учебным планом подготовки бакалавров по направлению 09.03.04 «Программная инженерия» Национального исследовательского университета «Высшая школа экономики», факультет компьютерных наук, департамент программной инженерии.

2. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

2.1. Постановка задачи на разработку программы

Программа должна содержать функцию вычисления значения факториала числа, которое помещается в положительное целое размером в двойное машинное слово, и поиск наибольшего числа, факториал которого удовлетворяет вышеописанному условию.

2.2. Описание алгоритма и функционирования программы

Программа содержит процедуру `factorialProc`, вычисляющую значение факториала числа, переданного в качестве параметра через стек. Результат возвращается через регистр `eax`. Если были переданы некорректные данные, то в регистре будет записано число -1 (в условиях программы некорректные данные не могут быть переданы, но сам метод должен уметь обрабатывать подобные данные). Если же факториал числа не помещается в положительное целое размером в двойное машинное слово, то в регистре будет записано число 0. В противном случае, в регистре будет записан факториал числа.

После запуска программы запускается цикл со счётчиком от 1 до того момента, как факториал счётчика перестанет помещаться в положительное целое размером в двойное машинное слово. До тех пор, пока факториал счётчика помещается, в консоль будет выводиться строка “<n>! = <n!>”, где n – счётчик. Как только факториал счётчика перестанет помещаться, в консоль будет выведена строка “<n>! Is too huge for double machine word” и программа будет завершаться после нажатия любой клавиши.

2.3. Описание и обоснование выбора метода организации входных и выходных данных

Входные данные для решения данной задачи не требуются. Вывод данных производится в консоль. Для наглядности выводятся числа, факториалы которых помещаются в положительное целое размером в двойное машинное слово, и сами факториалы. В конце выводится первое положительное число, факториал которого не помещается в положительное целое размером в двойное машинное слово.

3. КОД ПРОГРАММЫ

format PE console

entry start

include 'win32a.inc'

section '.data' data readable writable

nFactorialStr db '%d! = %d', 10, 0

nIsTooHugeStr db '%d! is too huge for double machine word', 10, 0

n dd 1

section '.code' code readable executable

; Начало программы (а также цикл по n).

start:

; Добавляем число, факториал которого

; необходимо посчитать, в стек - в

; качестве параметра для процедуры.

push dword[n]

; Вызываем саму процедуру подсчёта

; факториала.

call factorialProc

; Процедура возвращает значение через

; регистр eax. Если данные некорректны

; (что невозможно данных условиях), то

; вернётся -1. Если переполнение, то 0.

; В противном случае факториал числа.

; В данной ситуации, достаточно сранить

; eax с 0.

cmp eax, 0

```
jne isSuitable
```

```
jmp isNotSuitable
```

```
; Метка вывода факториала числа (помещается в  
; двойное машинное слово).
```

```
isSuitable:
```

```
; Добавляем в стек значение факториала n,  
; число n и информативную строку.
```

```
push eax
```

```
push [n]
```

```
push nFactorialStr
```

```
; Выводим в консоль.
```

```
call [printf]
```

```
; Очищаем стек.
```

```
add esp, 12
```

```
; Увеличиваем значение n.
```

```
mov ebx, [n]
```

```
inc ebx
```

```
mov [n], ebx
```

```
; Возвращаемся к началу программы.
```

```
jmp start
```

```
; Метка вывода числа, факториал которого не  
; помещается в двойное машинное слово.
```

```
isNotSuitable:
```

```
; Добавляем в стек число n и  
; информативную строку.
```

```
push [n]
```

```
push nIsTooHugeStr
```

```
; Выводим в консоль.
```

```
call [printf]
```

```
; Очищаем стек.
```

```
add esp, 8
```

```
; Метка окончания программы.
```


finish:

```
call [getch]
push 0
call [ExitProcess]
```

```
; Процедура вычисления факториала числа.
; Принимает через стек 1 параметр - число n,
; факториал которого необходимо вычислить.
; Результат возвращается в регистре eax.
; Если результат не вмещается в двойное
; машинное слово, то в eax помещается 0.
; Если в качестве параметра передано
; некорректное значение, то в eax
; помещается -1.
```

factorialProc:

```
; Помещаем в стек регистр ebp, который
; принято использовать для обращения к
; параметрам внутри процедуры.
push ebp
; Копируем значение регистра esp в
; регистр ebp, тем самым запоминаем
; в нём положение вершины стека для
; возможности адресовать параметры
; относительно регистра ebp.
mov ebp, esp
; Передано число, меньшее 1
; (некорректные данные).
cmp dword[ebp+8], 0
jle wrongInput
; Обнуляем значение регистра ebx,
; в котором будем хранить значения
; результатов вычислений.
```

```
xor ebx, ebx
```

```
; Делаем его равным единице для
; последующего пошагового домножения.
```

```
inc ebx
```

```
; Цикл подсчёта факториала.
```

```
factorialLoop:
```

```
; Очищаем регистры eax и edx на каждой
; итерации цикла.
```

```
xor eax, eax
```

```
xor edx, edx
```

```
; Копируем значение числа, факториал
; которого нужно вычислить в регистр eax.
```

```
mov eax, dword[ebp+8]
```

```
; Домножаем регистр eax на сохранённый
; ранее результат.
```

```
mul ebx
```

```
; В результате работы mul верхняя часть
; результата сохраняется в регистр edx,
; а нижняя - в регистр eax. Значит, если
; регистр edx не равен 0, то значение
; факториала числа не помещается в
; двойное машинное слово. В таком случае
; возвращается 0.
```

```
cmp edx, 0
```

```
jne tooHugeNum
```

```
; Если же edx равен 0, то используем
; данный регистр для перезаписи данных.
```

```
mov edx, dword[ebp+8]
```

```
dec edx
```

```
mov dword[ebp+8], edx
```

```
; Если edx равен 0, то факториал посчитан.
```

```
cmp edx, 0
```

; В таком случае, заканчиваем работу.

je finishFactorialProc

; В противном случае, запоминаем в ebx

; текущее значение.

mov ebx, eax

; Прыгаем на метку цикла.

jmp factorialLoop

; Метка некорректных данных.

wrongInput:

; Обнуляем регистр eax.

xor eax, eax

; Превращаем eax в -1.

dec eax

; Прыгаем на шаг окончания работы.

jmp finishFactorialProc

; Метка переполнения.

tooHugeNum:

; Обнуляем eax, так как переполнение.

xor eax, eax

; Метка окончания работы.

finishFactorialProc:

; Убираем из стека регистр ebx.

pop ebx

; Удаляем из стека двойное машинное

; слово, поданное на вход.

ret 4

section '.idata' import data readable

library kernel, 'kernel32.dll',\

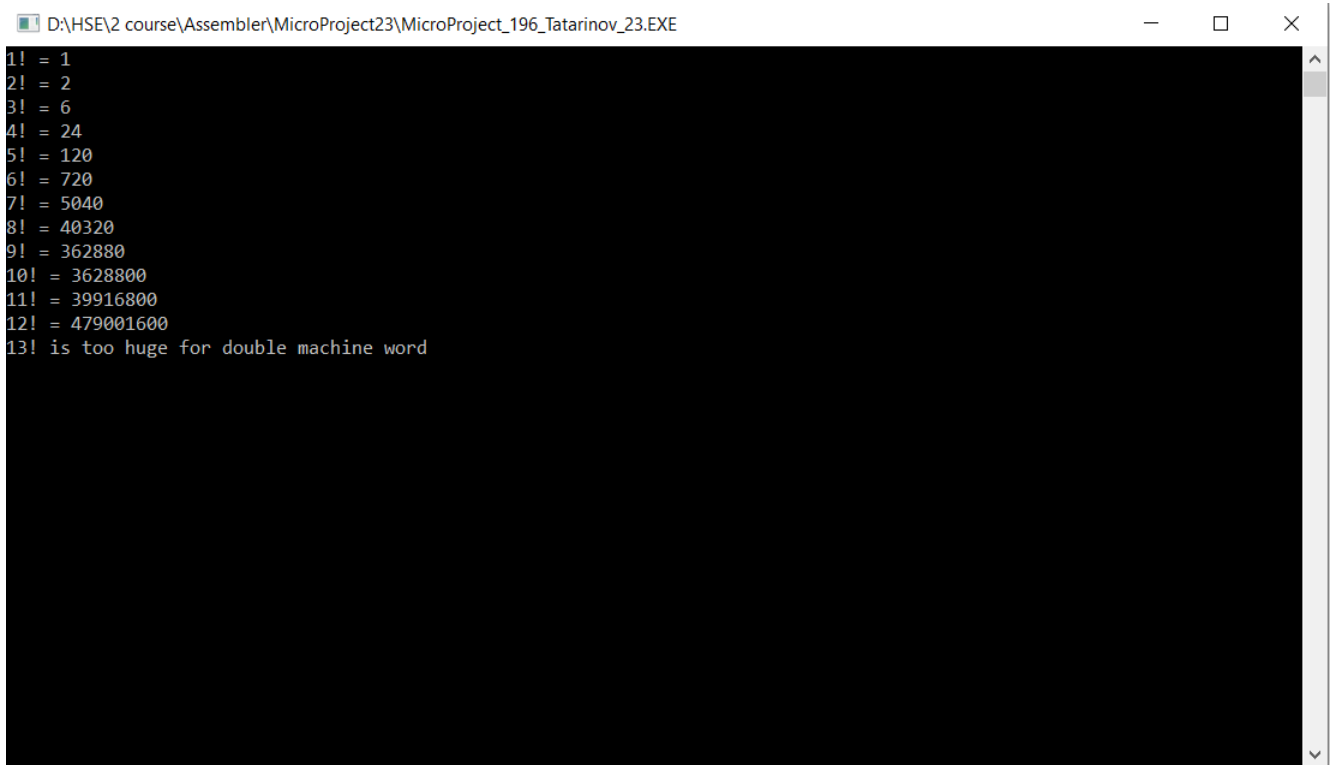
msvcrt, 'msvcrt.dll',\

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'
```

4. ТЕСТИРОВАНИЕ

После запуска программы выводится следующее:



The screenshot shows a Windows command prompt window titled "D:\HSE\2 course\Assembler\MicroProject23\MicroProject_196_Tatarinov_23.EXE". The window has a black background with white text. The text displays the factorial of numbers from 1 to 13. The output is as follows:

```
1! = 1
2! = 2
3! = 6
4! = 24
5! = 120
6! = 720
7! = 5040
8! = 40320
9! = 362880
10! = 3628800
11! = 39916800
12! = 479001600
13! is too huge for double machine word
```

После нажатия любой клавиши программа закрывается.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1) fasm – основное руководство [Электронные ресурс]// URL: <http://flatassembler.narod.ru/fasm.htm>
(режим доступа: свободный, дата обращения: октябрь 2020)
- 2) Учебный курс. Часть 25. Передача параметров через стек [Электронный ресурс]// URL:
<https://fasmworld.ru/uchebnyj-kurs/025-peredacha-parametrov-cherez-stek/> (режим доступа: свободный,
дата обращения: октябрь 2020)

ПРИЛОЖЕНИЕ 1

ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ ПЕРЕМЕННЫХ

Таблица 1.1

Описание и функциональное назначение переменных

Переменная	Назначение
nFactorialStr (строковая)	Строка-комментарий при выводе числа и его факториала
nIsTooHugeStr (строковая)	Строка-комментарий при выводе числа и информации о том, что его факториал не помещается в положительное целое размером в двойное машинное слово
n (числовая)	Счётчик, по которому вычисляется число, факториал которого не помещается в положительное целое размером в двойное машинное слово

ПРИЛОЖЕНИЕ 2
ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ ПРОЦЕДУР

Таблица 2.1

Описание и функциональное назначение процедур

Имя метода	Аргументы	Назначение
factorialProc	Число, факториал которого необходимо вычислить (передаётся через стек)	Вычисление факториала переданного в качестве параметра через стек числа