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

Факультет компьютерных наук

Департамент программной инженерии

Домашнее задание по дисциплине «Архитектура вычислительных систем»

Ибрагимов Константин Леонидович, БПИ192

Задача

Разработать программу, вычисляющую число не простых (составных) чисел в диапазоне от 1 до бинарного машинного слова без знака.

Реализация

Программа получает через механизм командной строки число N.

Далее проводится проверка, не превышает ли число размеров, при которых работа программы сильно затягивается. Если число слишком большое (>10^8) или отрицательное, в MessageBox выводится сообщение об этом (Рис. 1)

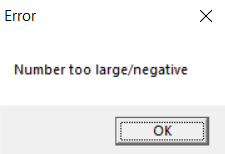


Рис.1 – вывод при некорректном аргументе

В противном случае программа итерируется по числам от 2 до введенного числа N, и проверяет, является ли число составным методом взятия остатка от деления N на числа от 2 до N-1.

Программа выводит количество составных чисел в MessageBox (Рис. 2)

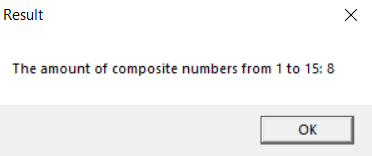


Рис.2 – вывод при корректном аргументе

Текст программы

format PE Console

include "..\INCLUDE\win32ax.inc" ;filepath to include

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;code section

.code

start:

cinvoke GetCommandLine ;get cl

mov [lpCommLine],eax ;save cl

cinvoke scanf, [lpCommLine], '%d', path, n ;read the number from cl

push n

push strScanInt

call [scanf]

cmp [n],100000000 ;checking input for too large/negative mumbers

ja .inputerror

;starting calculation

.countcomposite:

mov [count], 0 ;count = 0

mov [i], 2 ;i = 2

.fori:

mov eax, [i]

cmp eax,[n] ;if (i > A)

ja .finish ;break

;else

mov [j], 2 ;j = 2

.forj:

mov eax,[j] ;eax = j

imul eax, eax ;eax = j\*j

cmp eax,[i] ;if (j\*j > i)

ja .iloop ;break

;else

mov eax, [i] ;eax = i

cdq

idiv [j] ;edx = i % j

mov eax, edx ;eax = i % j

cmp eax, 0 ;if ( i % j != 0)

jne .jloop ;break

;else

inc [count] ;count++

jmp .iloop ;breakbreak

.jloop:

inc [j] ;j++

jmp .forj ;cycle

.iloop:

inc [i] ;i++

jmp .fori ;cycle

.finish:

;calculation end

cinvoke sprintf, res\_str, form\_str,[n],[count] ;froming output line

cinvoke MessageBox,0,res\_str,"Result",MB\_OK ;outputing the line

.close:

cinvoke ExitProcess,0 ;end

.inputerror:

cinvoke MessageBox,0,"Number too large/negative","Error",MB\_OK

;in this case the program will run for too much time

;so it is considered an error

jmp .close

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;data section

.data ; data readable writable

lpCommLine dd ? ;commanl line (cl)

form\_str db "The amount of composite numbers from 1 to %d: %d",0 ;output line format

res\_str db 512 dup(?) ;output line

path db 256 dup("?") ;cl address

n dd 1 ;input number

count dd 0 ;result

strScanInt db '%d', 0

i dd 0 ;outer cycle variable

j dd 0 ;inner cycle variable

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;import section

data import

library user32,'USER32.DLL', \

msvcrt,'MSVCRT.DLL', \

kernel32,'KERNEL32.DLL'

import kernel32, \

ExitProcess,'ExitProcess', \

GetCommandLine,'GetCommandLineA'

import user32, \

MessageBox, 'MessageBoxA'

import msvcrt, \

sprintf, 'sprintf', \

scanf, 'scanf'

end data