**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ**

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ**

**«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук

Образовательная программа бакалавриата «Программная инженерия»

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | Подп. и дата |  | | Инв. № дубл. |  | | Взам. инв. № |  | | Подп. и дата |  | | Инв. № подл. |  | | **Архитектура вычислительных систем**  **Микропроект №1, вариант 4**  **ПРОГРАММА ДЛЯ ОПРЕДЕЛЕНИЯ ВЗАИМНОЙ ПРОСТОТЫ ЧИСЕЛ**  **Пояснительная записка** | | |
|  |  | |
| Исполнитель:  студент группы БПИ191  В.И. Беловицкий/  2020 г. | |
|  | | |
|  | |  |

**СОДЕРЖАНИЕ**

1. ПОСТАВЛЕННАЯ ЗАДАЧА
   1. Задание

Разработать программу, решающую вопрос - являются ли четыре заданных числа взаимно простыми (числа задать машинными словами без знака).

* 1. Дополнительный функционал

Была добавлена возможность вводить количество чисел. Доступный диапазон ограничен.

1. ОБЩИЙ АЛГОРИТМ РАБОТЫ

Программа выполняется по следующему алгоритму:

* Ввод количества чисел
* Повтор ввода при некорректном вводе
* Последовательный ввод чисел
* Повтор ввода при некорректном вводе
* Последовательное выполнение алгоритма Евклида для введенных чисел
* Вывод результата – НОД чисел + являются ли они взаимно простыми

1. ТЕКСТ ПРОГРАММЫ

format PE console

entry start

include 'win32a.inc'

section '.data' data readable writable

strVecSize db 'Input how many numbers check on coprime: ', 0

strIncorNumber db 'Number should be in range [%hu, %hu]. Your input: %hu. Try again:',10, 0

strVecElemI db '[%hu]? ', 0

strScanNum db '%hu', 0

strNotCoprime db 'Numbers are not coprime.', 0

strAreCoprime db 'Numbers are coprime.', 0

strGCD db 'GCD = %hu', 0

strEndl db '', 10, 0

strPressKey db 'Press any key to exit...', 10, 0

vec\_size dd 0

vec\_min\_size dd 1

vec\_max\_size dd 10

num\_min dd 1

num\_max dd 65535

i dd ?

tmp dd ?

tmpStack dd ?

vec dd 0

;--------------------------------------------------------------------------

section '.code' code readable executable

start:

call VectorInput

push strEndl

call [printf]

call FindGCD

mov [tmp], eax

push eax

push strGCD

call [printf]

push strEndl

call [printf]

cmp [tmp], 1 ; GCD = 1 -> coprime

je AreCoprime

push strNotCoprime

call [printf]

jmp finish

AreCoprime:

push strAreCoprime

call [printf]

finish:

push strEndl

call [printf]

push strPressKey

call [printf]

call [getch]

push 0

call [ExitProcess]

;--------------------------------------------------------------------------

VectorInput:

mov [tmpStack], esp

push strVecSize

call [printf]

add esp, 4

push vec\_size

push strScanNum

call [scanf]

add esp, 8

mov eax, [vec\_size]

cmp eax, 0

jg checkUpperBound

jle sizeIncorrect

checkUpperBound:

mov eax, [vec\_size]

cmp eax, [vec\_max\_size]

jg sizeIncorrect

jle getVector

sizeIncorrect:

push [vec\_size]

push [vec\_max\_size]

push [vec\_min\_size]

push strIncorNumber

call [printf]

push strEndl

call [printf]

mov esp, [tmpStack]

jle VectorInput

getVector:

xor ecx, ecx

mov ebx, vec

getVecLoop:

mov [tmp], ebx

cmp ecx, [vec\_size]

jge endInputVector

inputNumber:

mov [i], ecx

push ecx

push strVecElemI

call [printf]

add esp, 8

push ebx

push strScanNum

call [scanf]

add esp, 8

mov eax, [ebx]

cmp eax, [num\_max]

jg incorNumber

cmp eax, [num\_min]

jl incorNumber

jmp correctNumber

incorNumber:

push eax

push [num\_max]

push [num\_min]

push strIncorNumber

call [printf]

add esp, 16

mov ecx, [i]

mov ebx, [tmp]

jmp inputNumber

correctNumber:

mov ecx, [i]

inc ecx

mov ebx, [tmp]

add ebx, 4

jmp getVecLoop

endInputVector:

ret

FindGCD:

xor ecx, ecx

mov ebx, vec

mov eax, [ebx]

GCDloop:

add ebx, 4

mov edx, [ebx]

inc ecx

gcd:

cmp eax, edx

jge fGEs

xchg eax, edx

fGEs:

cmp edx, 0

je gcdEnd

sub eax, edx

jmp gcd

gcdEnd:

cmp ecx, [vec\_size]

jge GCDFound

jmp GCDloop

GCDFound:

ret

;--------------------------------------------------------------------------

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'

include 'api\kernel32.inc'

import msvcrt,\

printf, 'printf',\

scanf, 'scanf',\

getch, '\_getch'

1. ВХОДНЫЕ ДАННЫЕ
   1. Ввод количества чисел

При запуске программы пользователю предлагается ввести количество чисел, которые будут проверяться на взаимную простоту.

1. ИСПОЛЬЗОВАННЫЕ АЛГОРИТМЫ
2. ТЕСТИРОВАНИЕ
3. ИСТОЧНИКИ