

Задание 5

5.1.22. Пусть есть описания

X dd ?; без знака

Y dd ?; без знака

Выписать последовательность не более 4-х команд реализующую оператор

$Y := 10 * Y + 9 * X$

Команды умножения не использовать. Считать, что значение выражения умещается в двойное слово.

Mov eax, x

Add eax, y

Lea eax, [8*eax+eax]

Add y, eax

5.2.22. Пусть есть описания

A dd ?; без знака

B dd ?; без знака

Выписать последовательность не более 4-х команд реализующую оператор

$B := 6 * B + 5 * A$

Команды умножения не использовать. Считать, что значение выражения умещается в двойное слово.

Mov eax, A

Add eax, B

Lea eax, [4*eax+eax]

Add B, eax

5.3. Пусть есть описание

A db ? ; без знака

B dw ? ; без знака

Выписать последовательность команд, реализующую оператор

$B := (A+B) * 3 \bmod 25$. Результат должен вычисляться правильно для любых значений A и B.

include console.inc

.data

.code

Start:

 inint eax; ввод A

 inint ebx; ввод B

 mov ecx, 3

 add eax, ebx

 mul ecx

 mov ecx, 25

 div ecx

 outint edx

Fin:

 exit

end Start

5.4. Пусть есть описание

A db ? ; со знаком

X dd 30 dup (?)

Y db 30 dup (?)

Выписать справа фрагмент программы (не более 6 команд) для преобразования массивов по правилу

$X[i] := X[i] + A$; $Y[i] := Y[i] + A$

.data

N equ 30

A db ?

x dd N dup(?), 0

y db N dup(?), 0

.code

Start:

mov ecx, N

movsx eax, A

mov dl, A

L: add x[4*ecx-4], eax

add y[ecx-1], dl

Loop L

5.5.

Пусть есть описание

B db ? ; со знаком

X dd 40 dup (?)

Y db 40 dup (?)

Выписать справа фрагмент программы (не более 6 команд) для преобразования массивов по правилу

$X[i] := X[i] - B$; $Y[i] := Y[i] - B$

.data

N equ 40

B db ?

x dd N dup(?), 0

y db N dup(?), 0

.code

Start:

mov ecx, N

movsx eax, B

mov dl, B

L: sub x[4*ecx-4], eax

sub y[ecx-1], dl

Loop L

5.6.

Пусть есть строки на Ассемблере

N = 40

X dd 40 dup (?) ; X[1..N]

Y db 40 dup (?) ; Y[1..N] со знаком

Выписать справа фрагмент программы (не более 5 команд) для преобразования массива X по правилу

$X[i] := X[i] - Y[i] - i$

.data

N equ 40

x dd N dup(?), 0

y db N dup(?), 0

.code

Start:

mov ecx, N

L: movsx eax, y[ecx-1]

sub x[4*ecx-4], eax

sub x[4*ecx-4], ecx

Loop L

5.7. Задание

Пусть есть описание

`X dd 80 dup (?) ; X[1..80]`

`Y db 80 dup (?) ; Y[0..79]`

Выписать справа фрагмент программы (не более 6 команд) для преобразования массивов по правилу

$X[i] := X[i] - i; Y[i] := Y[i] - i$

`.data`

`N equ 80`

`x dd N dup(?), 0`

`y db N dup(?), 0`

`.code`

Start:

`mov ecx, N`

`mov dl, 1`

`L: sub x[4*ecx-4], ecx`

`sub y[ecx-1], cl`

`add y[ecx-1], dl`

`Loop L`

5.8.

X T ?; число со знаком

Здесь T **equ db**, при чётном №, T **equ dw**, при нечётном №, где № – Ваш номер в ведомости. Выписать справа фрагмент на Ассемблере для вычисления и печати значения выражения

$(X*1000) \text{ div } 3$

.data

T equ dw;db

x T 1

.code

Start:

movsx eax, x

mov ebx, 1000

mul ebx ;<edx:eax>

mov ebx, 3

cdq

idiv ebx

outi eax

5.7. Выписать справа последовательность команд (не более 6-ти) для решения следующей задачи. Скопировать содержимое регистра AL в регистр BL, сдвинув все единицы к левому краю (для чётного №) или к правому краю (для нечётного №), где № – Ваш номер в ведомости. Регистры не портить. Например, для AL=01001010b и №=2 должно быть BL=11100000b, а для №=1 должно быть BL=00000111b.

xor ebx, ebx

mov ecx, 8

L: shr al, 1

jnc M

lea ebx, [ebx*2+1] | M: Loop L