**Вариант 2.1. Найти целое значение аргумента, при котором функция**

Y = 20000 / (8 \* x2 + 25) станет меньше 20.

Void main(){

long x=0;

\_asm{

M1: INC x

MOV eax,8

IMUL x

IMUL x

ADD eax,25

MOV esi,eax

MOV eax,20000

CDQ

IDIV esi

CMP eax,20

JNC M1

}

}  
  
  
**Вариант 3.1. В памяти задан массив из 10-ти элементов. Поместить в регистр EAX минимальный элемент массива, а в регистр EDX его адрес в памяти.**

Void main(){

long mas[10]={228,4,1337,1488,23,15,7,7,7,666};

\_asm{

LEA ebx,mas

MOV ecx,9

MOV eax, dword ptr[ebx]

MOV edx, ebx

M1: ADD ebx,4

CMP eax, dword ptr[ebx]

JC M2

MOV eax, dword ptr[ebx]

MOV edx, ebx

M2: LOOP M1

}

}

**Задача 1.3**

**Вариант 3. Вычислить 8 значений функции: Y = 5 \* x2 + 2 \* x – 14 (х - изменяется от 2 с шагом 4). Результат разместить в памяти.**

N = 8

Y = (5 \* x + 2) \* x – 14

X = 2

S = 4

void main() {

long X = 2, Y[8];

\_asm {

LEA EBX,Y

MOV ECX,8

M1: MOV EAX,5

IMUL X

ADD EAX,2

IMUL X

SUB EAX,14

MOV dword ptr[EBX],EAX

ADD EBX,4

ADD X,4

LOOP M1

}

}

**Задача 2.3**

**Вариант 3. Найти целое значение аргумента, при котором функция Y = 15 \* x2 + 11 \* x – 16 станет больше 2000.**

Y = (15 \* X + 11) \* X – 16

void main() {

long X = 0;

\_asm {

M1: INC X

MOV EAX,15

IMUL X

ADD EAX,11

IMUL X

SUB EAX,16

CMP EAX,2000

JC M1

}

}

**Задача 3.3.**

**Рассчитать и сохранить в памяти элементы массива, заданные функцией: Y = n! (для n от 1 до 8)**

void main() {

long mas[8];

\_asm {

LEA EBX, mas

MOV ECX, 8

MOV EAX, 1

MOV ESI, 1

M1: MUL ESI

MOV dword ptr[EBX], EAX

ADD EBX, 4

INC ESI

LOOP M1

}

}

**1.Обчислити 8 значень функції :**

**Y = (8 \* х2 + 12 \* х - 7) / (3 \* x + 25), (x - змінюється від 2 з кроком 3)**

void main () // начало программы на языке С++

{

long X=2; Y[8];

\_asm{ ; начало ассемблерной вставки

lea EBX,Y

mov ECX, 8

m1: mov EAX, 3

mul X

add EAX, 25

mov EDI, EAX

mov EAX, 8

imul X

add EAX, 12

imul X

sub EAX, 7

cdq

idw EDI, 1

cmp EDI, EDX

adc EAX, 0

mov dword ptr[EBX], EAX

add EBX, 4 ; увеличение адреса результатов

add X, 3 ; увеличение аргумента

loop m1 ; зацикливание по счетчику в ЕСХ

} // окончание ассемблерной вставки

}

**2. Визначити номер (n) елемента прогресії :**

**an = 5n + 8 \* n, при якому сума елементів прогресії перевищить 20000.**

void main () // начало программы на языке С++

{

long N=0; // ячейка памяти для аргумента

\_asm{ ; начало ассемблерной вставки

mov ESI, 1

mov EDI, 0

m1: inc N ; увеличение аргумента

mov EAX, 5 ; EAX = 5

mul P

mov P, EAX

add S, EAX ; накопление суммы

mov EAX, 8

mul N

add EAX, ESI ; накопление суммы

add EDI, EAX

cmp EDI, 20000

jc m1 ; переход, если сумма меньше 20000

} // окончание ассемблерной вставки

}

**В памяти задан массив из 10-ти элементов. Сохранить в регистре ESI количество нечетных элементов.**

void main(){

long y[10] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 }

\_asm{

Lea ebx,y

Mov ecx, 10

Mov esi, 0

M1: Shr dword ptr[ebx], 1

Adc esi,0

Loop M1

}

}

**БРИГАДА№9**

**Вариант 9.** Вычислить 6 значений функции:

***Y = (6x + 12) /(4\* x2 – 3)*** (***х*** - изменяется от 1 с шагом 1). Результат

округлить до целого и разместить в памяти.

Void main () {

Long x = 1, y[6];

\_asm {

Lea ebx, y; Адрес массива

Mov ecx, 6

Mov esi ,1

M1: Mov eax, x

Imul x

Add eax, eax;

Add eax, eax; Умножение на 4

sub eax, 3

Mov edi, eax

Mov eax, 6

Mul esi

Mov esi,eax;

Add eax,12

Cdq

Idiv edi

Shr edi, 1

Cmp edi, edx

Adc eax, 0

Mov dword ptr[ebx], eax

Add ebx, 4;

Inc x;

Loop M1

}

}

Y1 = 18 = 12hex

Y2 = 4 = 4hex

Y3 = 7 = 7hex

Y4 = 21 = 15hex

Y5 = 80 = 50hex

Y6 = 331 = 01 4b hex

*Вариант 9*

Найти целое значение аргумента, при котором функция Y = 300 \* х / (8^ X + 14) станет меньше 5.

void main(){

long x = 0;

\_asm{

Mov esi,1;

M1: inc x;

Mov eax , 8 ;

Imul esi;

Mov esi , eax;

Add eax,14;

Mov edi , eax;

Mov eax,300;

iMul x;

cdq

idiv edi;

cmp eax,5;

jnc M1;

}

}

При х = 3 у = 1.

# Задача 3

*Вариант 9*

В памяти задан массив из 10-ти элементов. Сохранить в регистре ESI количество единичных битов во всех элементах.

void main(){

long y[10] = { 1, 2, 3, 4, 5, 6, 7, 9, 10 }

\_asm{

Lea ebx,y;

Mov ecx,10

Mov esi ,0;

M1: mov eax, dword ptr[ebx];

Mov edx, 32;

M2: shr eax, 1;

Adc esi,0;

Dec edx;

Jnz M2;

Add ebx,4;

Loop M1;

}

}

ESI = 16 = 10hex

**1.5.** Вычислить 6 значений функции:

***Y = (2500 \* х – 8) / (3 \* х2 + 20*)** (***х*** - изменяется от 4 с шагом 3). Результат округлить до целого и разместить в памяти.

void main() {

long x=4, y[6];

\_asm{

Lea ebx,y;

Mov ecx,6;

M1: mov eax,3

iMul x

imul x

add eax,20

mov edi,eax

mov eax,2500

imul x

sub eax,8

cdq

idiv edi

shr edi,1

cmp edi,edx

adc eax,0

mov dword ptr[ebx],eax

add ebx,4

add x,5

loop M1

}

}

**2.5**. Найти целое значение аргумента, при котором функция

***Y = (2000 + х) / (8 \* x2 + 25)*** станет меньше 10.

void main()

{

long x=0;

\_asm

{

M1: INC x

MOV eax,8

IMUL x

IMUL x

ADD eax,25

MOV esi,eax

MOV eax,2000

ADD eax,x

CDQ

IDIV esi

CMP eax,10

JNC M1

}

}

**3.5**. В памяти задан массив из 8-ми элементов. Поместить в регистр EAX максимальный элемент массива, а в регистр ESI его адрес в памяти.

void main ()

{

long mas[8]={4,7,13,74,12,53,41,9};

\_asm

{

Lea ebx,mas

Mov ecx,7

Mov eax,dword ptr[ebx]

Mov esi,ebx

M1: add ebx,4

Cmp eax,dword ptr[ebx]

Jc M2

Mov eax,dword ptr[ebx]

Mov esi,ebx

M2: loop M1

nop

}

}