**Задание № 1.** Вычислить 7 значений функции:

***Y = (15 \* х2 – 12) / (4 \* x + 5)*** (***х*** - изменяется от 3 с шагом 3). Результат округлить до целого и разместить в памяти.

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

{

long X=3; // ячейка памяти для аргумента

long REZ[7]; // 7 ячеек памяти для результатов

\_asm{ ; начало ассемблерной вставки

lea EBX, REZ ; загрузка адреса результатов в регистр EBX

mov ECX, 7 ; счетчик количества повторений цикла

m1: mov EAX, 4 ; EAX = 4

imul X ; EAX = 4 \* x

add EAX, 5 ; EAX = 4 \* x + 5

mov EDI, EAX ; пересылка знаменателя в регистр EDI

mov EAX, 15 ; EAX = 15

imul X ; EAX = 15 \* x

imul X ; EAX = 15 \* x2

sub EAX, 12 ; EAX = 15 \* x2 – 12

cdq ; расширение операнда-делимого в EAX-EDX

div EDI ; частное – EAX , остаток – EDX

shr EDI, 1 ; деление знаменателя (делителя) на 2

cmp EDI, EDX ; сравнение половины делителя с остатком

adc EAX, 0 ; добавление к частному заема от сравнения

mov dword ptr[EBX], EAX ; пересылка результата в память

add EBX, 4 ; увеличение адреса результатов

add X, 3 ; увеличение аргумента

loop m1 ; зацикливание по счетчику в ЕСХ

} // окончание ассемблерной вставки

}

**Задание № 1.** 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 () // начало программы на языке С++

{

long X=4; // ячейка памяти для аргумента

long REZ[6]; // 6 ячеек памяти для результатов

\_asm{ ; начало ассемблерной вставки

lea EBX, REZ ; загрузка адреса результатов в регистр EBX

mov ECX, 6 ; счетчик количества повторений цикла

m1: mov EAX, 3 ; EAX = 3

imul X ; EAX = 3 \* x

imul X ; EAX = 3 \* x2

add EAX, 20 ; EAX = 3 \* x2 + 20

mov EDI, EAX ; пересылка знаменателя в регистр EDI

mov EAX, 2500 ; EAX = 2500

imul X ; EAX = 2500 \* x

sub EAX, 8 ; EAX = 2500 \* x – 8

cdq ; расширение операнда-делимого в EAX-EDX

div EDI ; частное – EAX , остаток – EDX

shr EDI, 1 ; деление знаменателя (делителя) на 2

cmp EDI, EDX ; сравнение половины делителя с остатком

adc EAX, 0 ; добавление к частному заема от сравнения

mov dword ptr[EBX], EAX ; пересылка результата в память

add EBX, 5 ; увеличение адреса результатов

add X, 3 ; увеличение аргумента

loop m1 ; зацикливание по счетчику в ЕСХ

} // окончание ассемблерной вставки

}

**Задание № 2.** Определить номер (*n*) элемента прогрессии :

***an = 8n – 5 \* n***, при котором сумма элементов прогрессии превысит 10000.

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

{

long N=0; // ячейка памяти для аргумента

long S=0; // ячейка для хранения суммы

long P=1; // ячейка для накопления 8n

\_asm{ ; начало ассемблерной вставки

m1: inc N ; увеличение аргумента

mov EAX, 8 ; EAX = 8

mul P ; умножение – 8n

mov P, EAX ; пересылка 8n в ячейку памяти

add S, EAX ; накапливание суммы

mov EAX, 5 ; EAX = 5

mul N ; EAX = 5 \* n

sub S, EAX ; накапливание суммы

cmp S, 10000 ; сравнение суммы с 10000

jc m1 ; переход, если сумма меньше 10000

} // окончание ассемблерной вставки

}

**Задание № 2.** Найти целое значение аргумента, при котором функция

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

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

{

long X=0; // ячейка памяти для аргумента

long Y=0; // ячейка для хранения суммы

\_asm{ ; начало ассемблерной вставки

m1: mov EAX, 8 ; EAX = 8

imul X ; EAX = 8 \* x

imul X ; EAX = 8 \* x2

add EAX, 25 ; EAX = 8 \* x2 + 25

mov EDI, EAX ; пересылка знаменателя в регистр EDI

mov EAX, 2000 ; EAX = 2000

add X ; EAX = 2000 + x

cdq ; расширение операнда-делимого в EAX-EDX

div EDI ; частное – EAX , остаток – EDX

mov dword ptr[EBX], EAX ; пересылка результата в память

add EBX, 5 ; увеличение адреса результатов

add X, 1 ; увеличение аргумента

loop m1 ; зацикливание по счетчику в ЕСХ

cmp [EBX], 10 ; сравнение функции с 10

jc m1 ; переход, если функция больше 10

} // окончание ассемблерной вставки

}

**Задание № 3.** В памяти задан массив из 5-ти элементов. Поместить в регистр EAX максимальный элемент массива, а в регистр EDI его адрес в памяти.

void main () {

long x[5]={23, 56, 84, 15, 74} // массив в памяти

\_asm { ; начало ассемблерной вставки

lea EBX, x ; начальный адрес массива – в ЕВХ

mov ECX, 4 ; счетчик повторений

mov EAX, dword ptr[EBX] ; первый элемент – в ЕАХ

m2: add EBX, 4 ; увеличение адреса

cmp EAX, dword ptr[EBX] ; сравнение со следующим элемент.

jc m1 ; переход, если меньше

mov EAX, dword ptr[EBX] ; больший элемент – в ЕАХ

mov EDI, EBX ; адрес элемента – в EDI

m1: loop m2 ; зацикливание по счетчику

}}

**Задание № 4.** В памяти задан массив из 8-ми элементов. Отсортировать элементы массива по возрастанию.

void main () {

long x[8]={7, 23, 56, 33, 84, 15, 11, 74};

\_asm {

mov EDX, 7 ; счетчик внешнего цикла - на 1 меньше

; количества элементов массива

m3: lea EBX, x ; начальный адрес массива

mov ECX, EDX ; счетчик внутреннего цикла

m2: mov EAX, dword ptr[EBX] ; элемент массива – в ЕАХ

add EBX, 4

cmp EAX, dword ptr[EBX] ; сравнение соседних элементов

jc m1 ; переход, если меньше

xchg dword ptr[EBX], EAX ; обмен элементов массива

mov dword ptr[EBX-4], EAX ;

m1: loop m2 ; окончание внутреннего цикла

dec EDX ; уменьшение счетчика внешнего цикла

jnz m3 ; окончание внешнего цикла

}}

**Вариант 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 x[8]={23, 56, 84, 15, 74, 32, 45, 61} // массив в памяти

\_asm { ; начало ассемблерной вставки

lea EBX, x ; начальный адрес массива – в ЕВХ

mov ECX, 7 ; счетчик повторений

mov EAX, dword ptr[EBX] ; первый элемент – в ЕАХ

m2: add EBX, 7 ; увеличение адреса

cmp EAX, dword ptr[EBX] ; сравнение со следующим элемент.

jc m1 ; переход, если меньше

mov EAX, dword ptr[EBX] ; больший элемент – в ЕАХ

mov EDI, EBX ; адрес элемента – в EDI

m1: loop m2 ; зацикливание по счетчику

}}

FPU

**Вариант 5.**  Вычислить 6 значений функции :

*Y = (20 \* x) /(5 \* x2 – 8,5)*  (*x* – изменяется от 1 с шагом 4). Результат разместить в памяти.

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

{

long A =5, B=20, x = 1;

float RES[6];

float C = 8.5;

float part;

\_asm{

lea EBX, RES ; загрузка адреса результатов в регистр EBX

mov ECX, 6 ; счетчик количества повторений цикла

m1: finit ; очистка регистров сопроцессора

fild x ; x

fimul x ; x2

fimul A ; 5x2

fsub C ; 5x2 – 8.5

fstp part ; сохранение результата в ячейке памяти

fild B ; 20

fimul x ; 20x

fdiv part ; деление

fstp RES

add EBX, 4

add x, 4

loop m1

}

}

**Задание №3.** Определить номер (*n*) элемента прогрессии :

***an = 5,3n + 5 \* n*,** при котором сумма элементов превысит 20000.

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

{

float A=5.3; // описание операндов в памяти

long B=5, C=20000, N=0;

\_\_asm{ ; начало ассемблерной вставки

finit ; очистка регистров сопроцессора

fld1 ; регистр для вычисления степенной функц.

fldz ; регистр для накопления суммы прогрессии

m1: inc N ;наращивание аргумента

fld A ; загрузка в ST(0) 5,3

fmulp ST(2),ST ; вычисление степенной функции 5,3n

fild B

fimul N

fadd ST,ST(2) ; вычисление элемента прогрессии

fadd ; накопление суммы прогрессии

ficom C ; сравнение суммы с 20000

fstsw AX ; сохранение регистра состояния SW (FPU) в

; регистре AX (CPU)

sahf ;сохранение старшего байта АХ в рег. флагов

jc m1 ; переход, если сумма меньше 20000

} // окончание ассемблерной вставки

} // окончание программы на языке С++

**Задание № 1.** Вычислить значение функции :



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

{

long A=20, B=30, M2=2; // описание операндов в памяти

float ALPHA=0.7, Y;

\_\_asm{ ; начало ассемблерной вставки

finit ; очистка регистров сопроцессора

fld ALPHA ; загрузка в вершину стека аргумента

fsin ; вычисление синуса

fimul A ; умножение вершины стека на константу

fimul B

fidiv M2

fstp Y ; сохранение результата в ячейке памяти

} // окончание ассемблерной вставки

} // окончание программы на языке С++

**Вариант 5.** Задан массив с элементами : *a(i) = sin (5 \* i).*

Определить номер элемента массива, при котором сумма элементов превысит 3. Аргумент синуса задан в градусах.

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

{

long A=5, C=3, i=0;

\_\_asm{ ; начало ассемблерной вставки

finit ; очистка регистров сопроцессора

fld1 ; регистр для вычисления функц.

fldz ; регистр для накопления суммы прогрессии

m1: inc i ;наращивание аргумента

fld A

fmul i

fsin ; вычисление синуса

fadd ST,ST(2) ; вычисление элемента прогрессии

fadd ; накопление суммы прогрессии

ficom C ; сравнение суммы с 3

fstsw AX ; сохранение регистра состояния SW (FPU) в

; регистре AX (CPU)

sahf ;сохранение старшего байта АХ в рег. флагов

jc m1 ; переход, если сумма меньше 3

} // окончание ассемблерной вставки

} // окончание программы на языке С++