

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
Національний аерокосмічний університет ім. М.Є. Жуковського  
«Харківський авіаційний інститут»

Факультет радіоелектроніки, комп'ютерних систем та інфокомунікацій

Кафедра комп'ютерних систем, мереж і кібербезпеки (503)

Лабораторна робота № 1

*Лінійні програми. Арифметичні команди Intel x86*

(назва лабораторної роботи)

з дисципліни

*Архітектура комп'ютерів*

(шифр)

ХАІ.503.525а.03О.123-Комп'ютерна інженерія, ПЗ №9629619

Виконав студент гр. 525а Литвиненко А.В.  
(№ групи) (П.І.Б.)

\_\_\_\_\_  
(підпис, дата)

Перевірів канд. техн. наук, доцент

В. І. Дужий  
(підпис, дата) (П.І.Б.)

Харків – 2022

## Варіант 12

### 1. Умова:

Операнди  $a$ ,  $b$  і  $c$  (за потреби), розміщені у пам'яті. При обчисленні значень арифметичних виразів використовують значення, обчислені з використанням значень операндів за формулами:  $a/(n+2)$ ,  $b/(n+3)$ ,  $c/(n+4)$ , де  $n$  - номер варіанта.

Значення лівої та правої частин арифметичного виразу: `left` та `right`.

### 2. Вхідні дані

$A, b, c$  – довгі цілі знакові числа

$2, 0$  – константи

### 3. Потрібний результат

`Left`, `right` – змінні, довгі знакові числа для фрагменту на `Ci`

`Left_a`, `right_a` – змінні, довгі цілі числа для фрагменту на асемблері

### 4. Алгоритм на псевдокоді

Вести значення змінних;

Вирахувати значення за формулами;

Вирахувати ліву частину на `Ci`;

Вирахувати праву частину на `Ci`;

Вирахувати на асемблері:

Занести нулі у змінні;

Підрахувати за формулою змінні;

Підрахувати ліву частину:

Підрахувати  $a+b$ ;

Підрахувати  $a+b+c$ ;

Підрахувати  $(a+b+c)**2$ ;

Підрахувати праву частину:

Підрахувати  $a*a$ ;

Підрахувати  $b*b$ ;

Підрахувати  $c*c$ ;

Підрахувати  $2*a*b$ ;

Підрахувати  $2*a*c$ ;

Підрахувати  $2*b*c$ ;

Підрахувати все вище;

Вивести на екран.

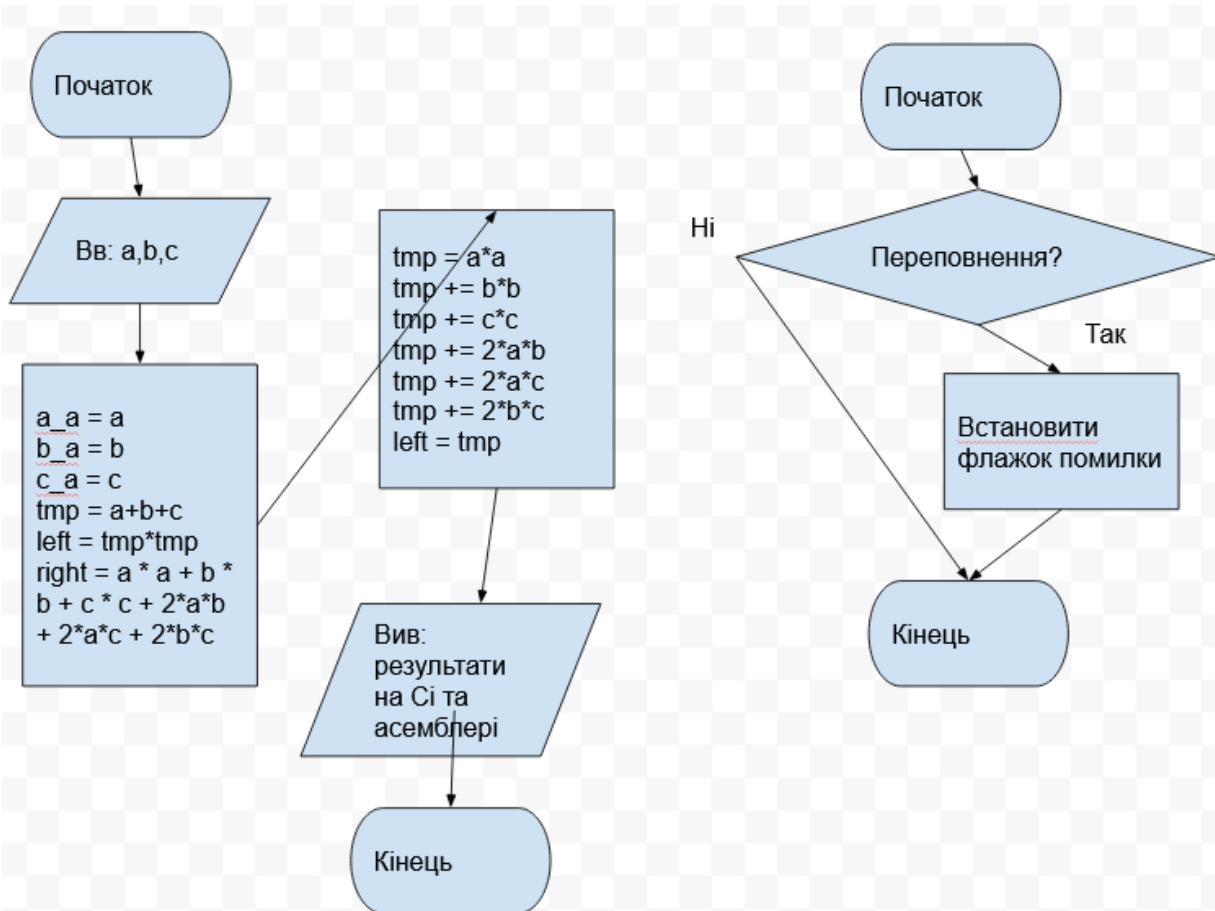


Рисунок 1 – Схема алгоритму

## 5. Текст програми

```

6. #include <stdio.h>
7. #include <intrin.h>
8.
9. long int
10.     tmp = 0, tmp2 = 0,
11.     right = 0, left = 0,
12.     right_a = 0, left_a = 0,
13.     a, b, c,
14.     a_a, b_a, c_a;
15. int err;
16. int err_la;
17.
18. #define VARIANT 12
19.
20. int main() {
21.     printf("Lytvynenko Andrii 525a\n");
22.     printf("The equation: (a+b+c)**2 = a**2 + b**2 + c**2 + +2*a*b + 2*a*c + 2*b*c\n");
23.
24.     for (;;) {
25.         printf("Please enter a,b,c: ");
26.         scanf("%li%li%li", &a, &b, &c);
27.
28.         a_a = a;
29.         b_a = b;
30.         c_a = c;
31.
32.         a = a / (VARIANT + 2);
33.         b = b / (VARIANT + 3);
34.         c = c / (VARIANT + 4);
  
```

```

35.
36. tmp = a + b + c;;
37. left = tmp * tmp;
38.
39. right = a * a + b * b + c * c + 2*a*b + 2*a*c + 2*b*c;
40.
41. err = 0;
42.
43. // (a+b+c)**2 = a**2 + b**2 + c**2 + +2*a*b + 2*a*c + 2*b*c
44. __asm {
45.     mov     ebx,0
46.     mov     right_a,ebx
47.
48.     mov     ebx,0
49.     mov     left_a,ebx
50.
51.     // a
52.     mov     eax,a_a
53.     mov     ebx,VARIANT
54.     add     ebx,2
55.     cdq
56.     idiv    ebx
57.     mov     a,eax
58.
59.     // b
60.     mov     eax,b_a
61.     mov     ebx,VARIANT
62.     add     ebx,2
63.     cdq
64.     idiv    ebx
65.     mov     b,eax
66.
67.     // c
68.     mov     eax,c_a
69.     mov     ebx,VARIANT
70.     add     ebx,2
71.     cdq
72.     idiv    ebx
73.     mov     c,eax
74.
75.     // left
76.     mov     ebx,a
77.     add     ebx,b    // a + b
78.     mov     eax,c
79.     add     ebx,eax  // a + b + c
80.     imul    ebx,ebx  // (a+b+c)**2
81.     mov     left_a,ebx // left_a = (a+b+c)**2
82.
83.     // right
84.     mov     ebx,a
85.     imul    ebx,ebx
86.     add     right_a,ebx
87.
88.     mov     ebx,b
89.     imul    ebx,ebx
90.     add     right_a,ebx
91.
92.     mov     ebx,c
93.     imul    ebx,ebx
94.     add     right_a,ebx
95.
96.     // (a+b+c)**2 = a**2 + b**2 + c**2 + +2*a*b + 2*a*c + 2*b*c
97.     mov     ebx,2
98.     imul    ebx,a
99.     imul    ebx,b

```

```

100.          add            right_a,ebx          // right_a = (a**2 + b**2 + c**2) + 2*a*b
101.
102.          mov            ebx,2
103.          imul           ebx,a
104.          imul           ebx,c
105.          add            right_a,ebx          // right_a = (a**2 + b**2 + c**2) + 2*a*b + 2*a*c
106.
107.          mov            ebx,2
108.          imul           ebx,b
109.          imul           ebx,c
110.          add            right_a,ebx          // right_a = (a**2 + b**2 + c**2) + 2*a*b + 2*a*c
+ 2*b*c
111.
112.          }
113.
114.          printf("A: %li\nB: %li\nC: %li\n", a,b,c);
115.
116.          printf("[C] Left: %li \t Right: %li\n", left, right);
117.          printf("[ASM] Left: %li \t Right: %li\n", left_a, right_a);
118.      }
119.
120.
121.      return 0;
122. }
123.

```

## 6. Тестові приклади

№	Вх. дані	Оч. рез.	От. рез.	Ціль
1	30 30 30	25	25	Додатні числа
2	-25 -32 -12	9	9	Від'ємні числа
3	9999 9999 9999	8982009	4016016 4588164	Переповнення

```
C:\Users\user\Desktop\UNIVER\Architecture\laba_1\Laba_01\Debug\Laba_01.exe
Lytvynenko Andrii 525a
The equation: (a+b+c)**2 = a**2 + b**2 + c**2 + +2*a*b + 2*a*c + 2*b*c
Please enter a,b,c: 30 30 30
A: 2
B: 2
C: 2
[C] Left: 25      Right: 25
[ASM] Left: 36    Right: 36
Please enter a,b,c: -25 -32 -12
A: -1
B: -2
C: 0
[C] Left: 9       Right: 9
[ASM] Left: 9     Right: 9
Please enter a,b,c: 9999 9999 9999
A: 714
B: 714
C: 714
[C] Left: 4016016  Right: 4016016
[ASM] Left: 4588164 Right: 4588164
Please enter a,b,c: _
```

Рисунок 2 – скриншот виконання

## Висновок

Під час цієї лабораторної роботи я вивчив найпростіші арифметичні операції на асемблері та навчився їх використовувати на практиці, написав свою першу програму та протестував її.