Правительство Российской Федерации
Федеральное государственное автономное образовательное учреждение
Высшего профессионального образования
Национальный исследовательский институт
«Высшая школа экономики»
Московский институт электроники и математики
Компьютерная безопасность

Отчет
по лабораторной работе №4.1
по курсу «Язык ассемблера»

Вариант №33

Ф.И.О. студента	Номер группы	Дата	Баллы
Николаев Александр Александрович	СКБ191	10.04.2022	

Задание А4

Москва, 2022 г.

Задача

Дан массив A из 16 байтов. Сосчитать сумму модулей тех элементов, величина которых превышает –8 (результат в слове). Скопировать эти элементы в массив B и сосчитать их количество. В массив C поместить адреса (смещения) этих элементов.

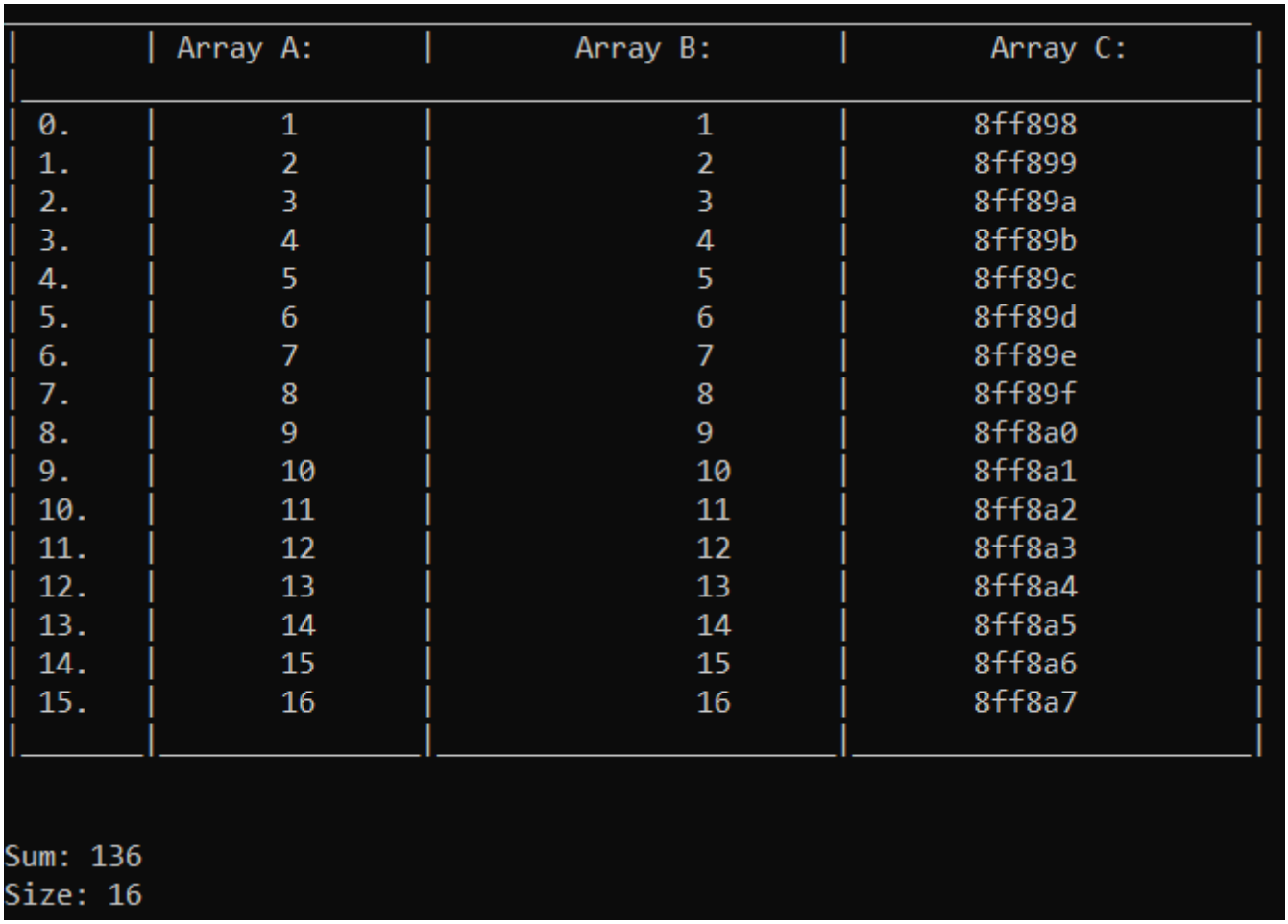
Текст программы

```
#include <stdio.h>
#include <string.h>
#include <limits.h>
#include <math.h>

int main() {
	char A[16];
	char B[16];
	int C[16];

	int item;

	short int siz = 0;
	short int sum = 0;

	memset(B, 0, sizeof(B));
	memset(C, 0, sizeof(C));

	for (int i = 0; i < 16; ++i) {
		while (1) {
			printf("Enter number #%d: ", i);
			scanf_s("%d", &item);
			if (abs(item) <= CHAR_MAX) {
				A[i] = item;
				break;
			}
			else {
				printf("Value out of range! Try again\n\n");
			}
		}
	}

	_asm {
		lea esi, A;			// загрузить исполнительный адрес массива A
		lea edi, B;			// загрузить исполнительный адрес массива B
		lea ebx, C;			// загрузить исполнительный адрес массива C

		mov ecx, 16;			// установить счётчик цикла

		mov ax, 0;			// под расширение чисел из массива A
		mov dx, 0;			// под сумму чисел из массива A

	BODY: cmp byte ptr [esi], -8;		// сравнить элемент массива A с числом -8
		jng NEXT;			// если элемент массива A меньше либо равен -8,
то перейти к следующей итерации

		inc ax;				// иначе увеличить значение счётчика
		push ax;			// поместить значение счётчика в стек
```

```
            mov al, [esi];                // скопировать значение элемента массива A в
регистр al (как байт)

            mov [edi], al;                // поместить в элемент массива B значение
подходящего элемента из массива A
            inc edi;                      // переместить указатель на следующий элемент
массива B

            mov [ebx], esi;               // поместить в элемент массива C адрес
подходящего элемента из массива A
            add ebx, 4;                   // переместить указатель на следующий элемент
массива C

            cbw;                          // знаковое расширение до слова
            cmp ax, 0;                    // сравнить число в регистре ax с нулём
            jge INCR;                     // если в регистре ax число положительно, то
перейти к сложению
            neg ax;                       // иначе взять модуль числа

      INCR: add dx, ax;                   // прибавить к сумме значение регистра ax
            pop ax;                       // восстановить значение счётчика из стека

      NEXT: inc esi;                      // перейти к следующему элементу массива
            dec ecx;                      // уменьшить значение счётчика цикла на 1
            cmp ecx, 0;                   // проверить значение счётчика
            jne BODY;                     // если не 0, то перейти к следующей итерации
            nop;                          // иначе закончить работу цикла

            mov sum, dx;                  // поместить в переменную sum значение суммы
модулей подходящих чисел
            mov siz, ax;                  // поместить в переменную siz
      }

      printf("_____________________________________________________________");
      printf("\n|   \t| Array A: \t|\t Array B: \t|\t Array C:\t|\n");
      printf("|_______|_______________|_____________________|_____________________|\n
");
      for (int i = 0; i < sizeof(A) / sizeof(short int); ++i) {
            printf("| %d. \t|\t%d", i, A[i]);
            if (i < siz) {
                  printf("\t|\t\t%d\t|\t%x\t\t|\n", B[i], C[i]);
            }
            else {
                  printf("\t| \t\t\t|\t\t\t|\n");
            }
      }
      printf("|_______|_______________|_____________________|_____________________|\n
");

      printf("\n\nSum: %d\nSize: %d\n\n", sum, siz);

      return 0;
}
```

Тестирование программы

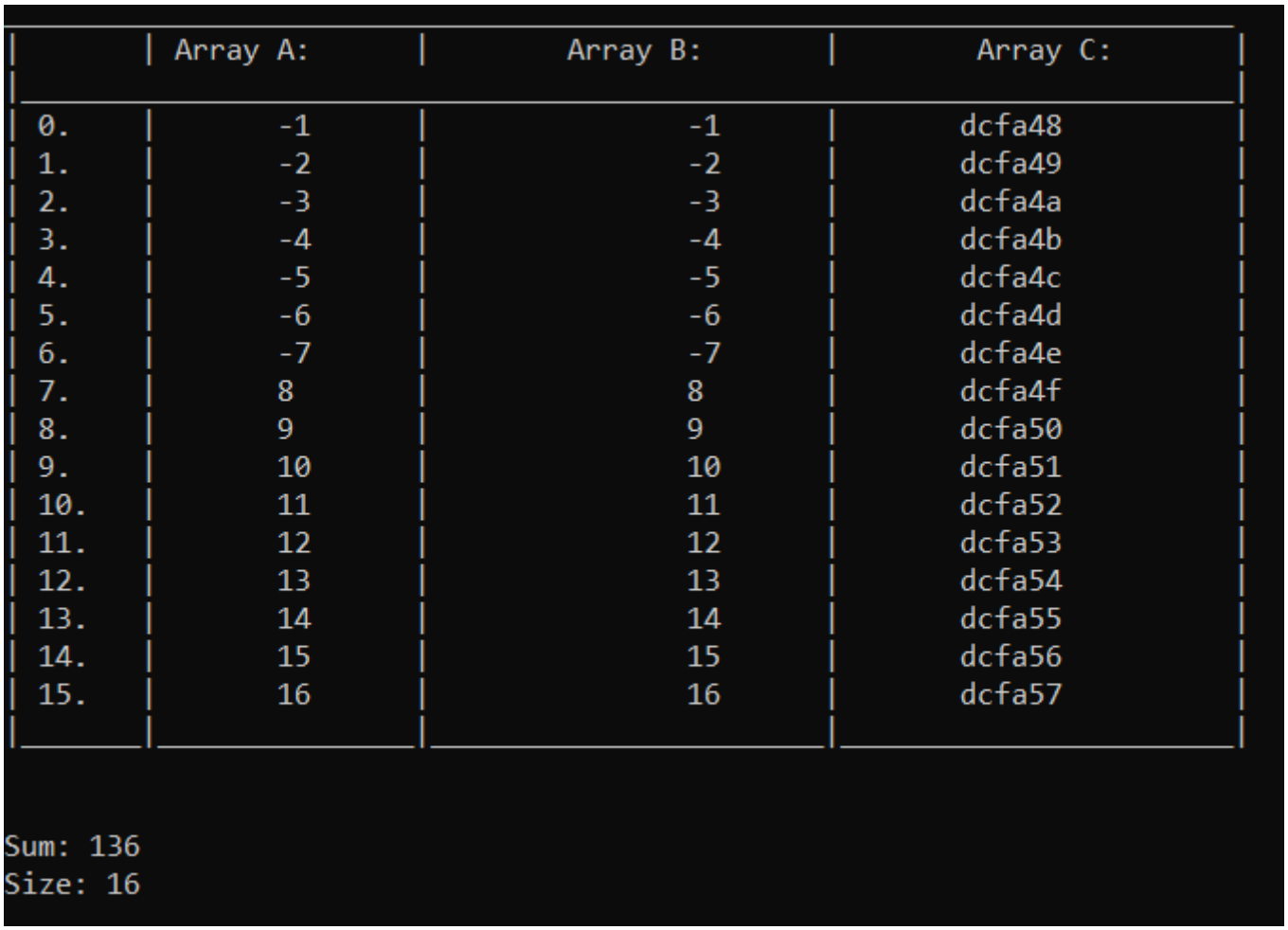
Все значения доступны для суммирования, сумма 136, количество допустимых значений – 16.



	Array A:	Array B:	Array C:
0.	1	1	8ff898
1.	2	2	8ff899
2.	3	3	8ff89a
3.	4	4	8ff89b
4.	5	5	8ff89c
5.	6	6	8ff89d
6.	7	7	8ff89e
7.	8	8	8ff89f
8.	9	9	8ff8a0
9.	10	10	8ff8a1
10.	11	11	8ff8a2
11.	12	12	8ff8a3
12.	13	13	8ff8a4
13.	14	14	8ff8a5
14.	15	15	8ff8a6
15.	16	16	8ff8a7

Sum: 136
Size: 16

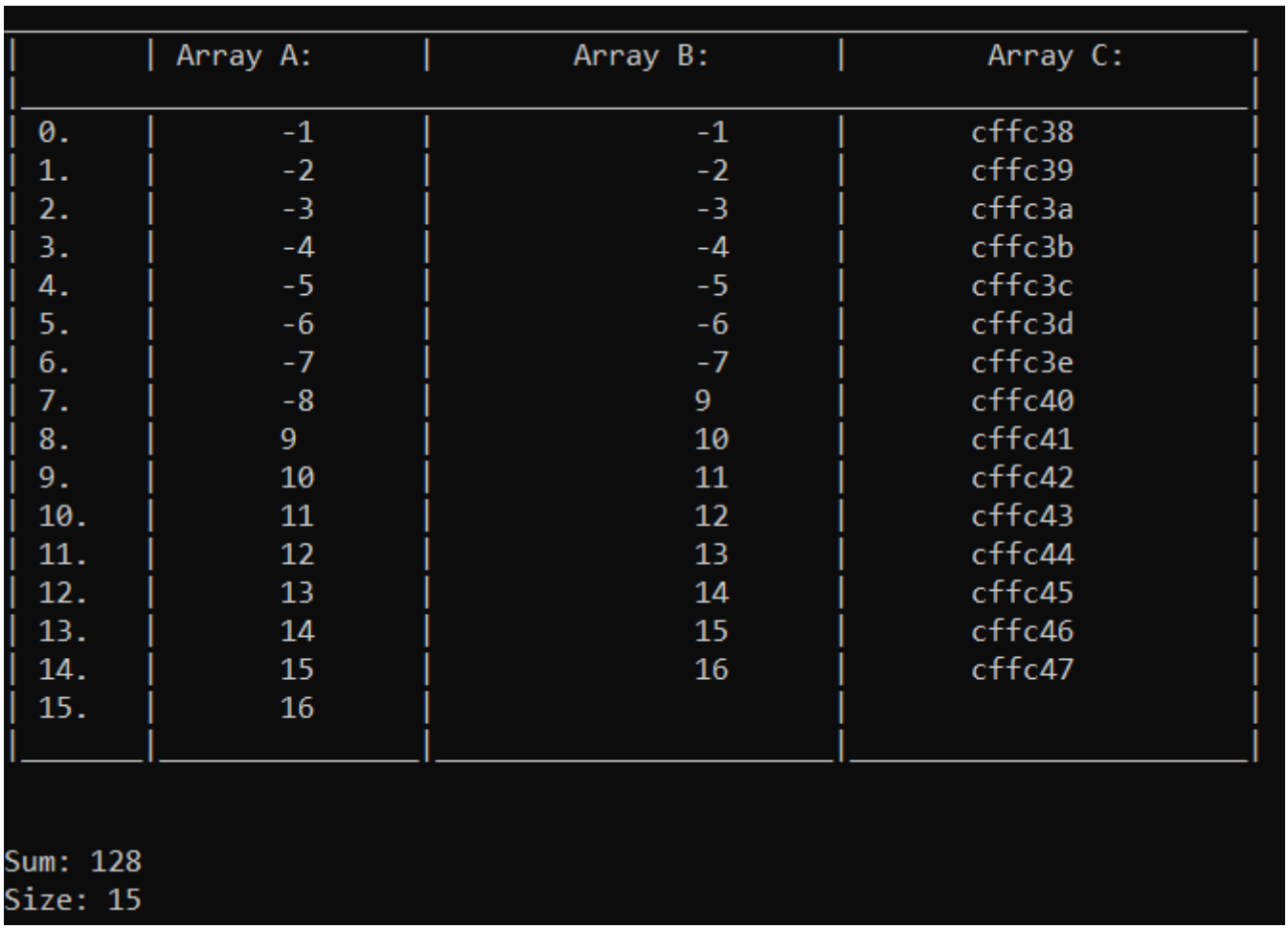
Все значения доступны для суммирования, сумма 136, количество допустимых значений – 16.



	Array A:	Array B:	Array C:
0.	-1	-1	dcfa48
1.	-2	-2	dcfa49
2.	-3	-3	dcfa4a
3.	-4	-4	dcfa4b
4.	-5	-5	dcfa4c
5.	-6	-6	dcfa4d
6.	-7	-7	dcfa4e
7.	8	8	dcfa4f
8.	9	9	dcfa50
9.	10	10	dcfa51
10.	11	11	dcfa52
11.	12	12	dcfa53
12.	13	13	dcfa54
13.	14	14	dcfa55
14.	15	15	dcfa56
15.	16	16	dcfa57

Sum: 136
Size: 16

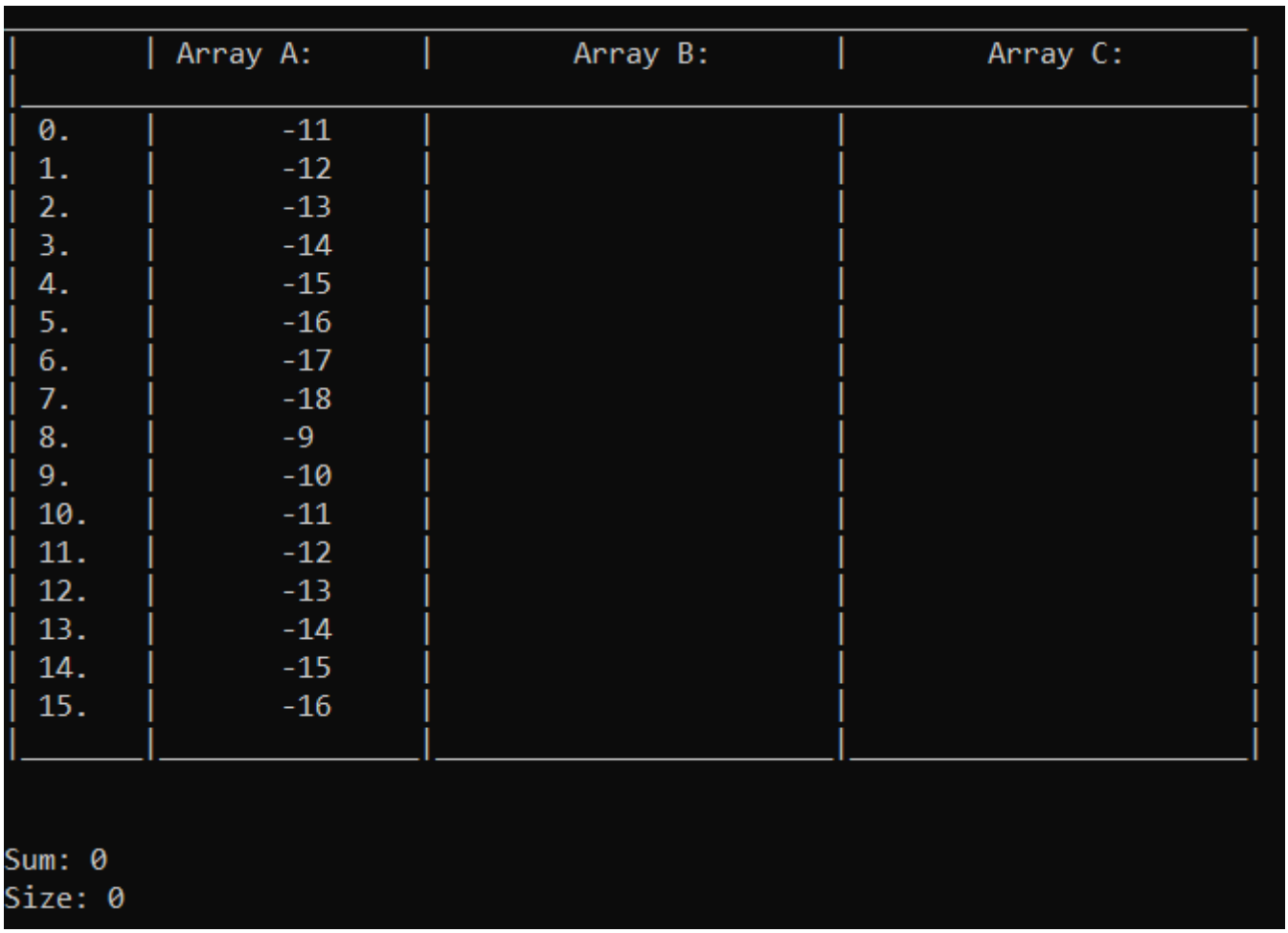
Доступны для суммирования все элементы, кроме -8, сумма 128, количество допустимых значений – 15.



	Array A:	Array B:	Array C:
0.	-1	-1	cffc38
1.	-2	-2	cffc39
2.	-3	-3	cffc3a
3.	-4	-4	cffc3b
4.	-5	-5	cffc3c
5.	-6	-6	cffc3d
6.	-7	-7	cffc3e
7.	-8	9	cffc40
8.	9	10	cffc41
9.	10	11	cffc42
10.	11	12	cffc43
11.	12	13	cffc44
12.	13	14	cffc45
13.	14	15	cffc46
14.	15	16	cffc47
15.	16		

Sum: 128
Size: 15

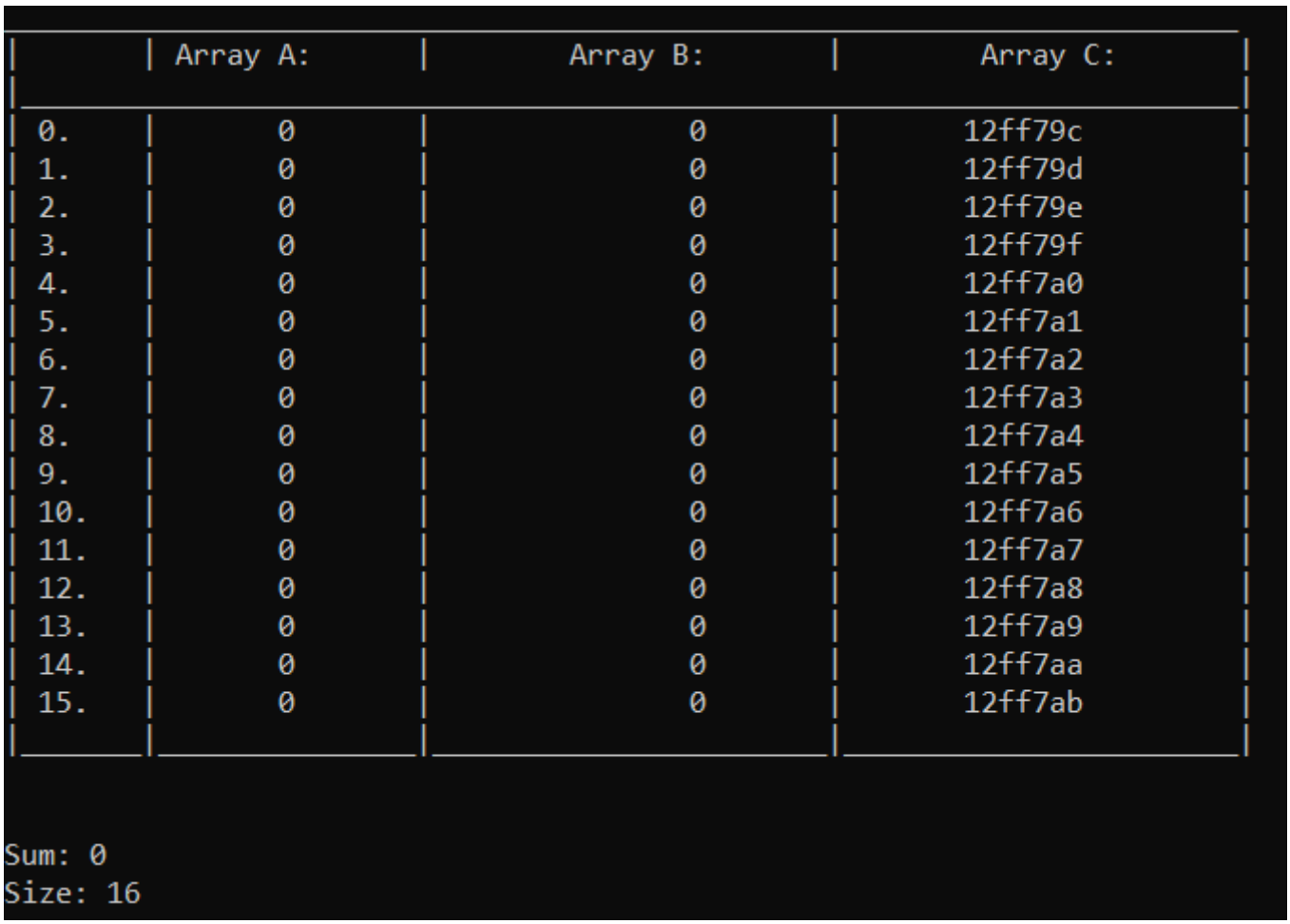
Ни одно значение недоступно для суммирования, сумма 0, количество допустимых значений – 0.



	Array A:	Array B:	Array C:
0.	-11		
1.	-12		
2.	-13		
3.	-14		
4.	-15		
5.	-16		
6.	-17		
7.	-18		
8.	-9		
9.	-10		
10.	-11		
11.	-12		
12.	-13		
13.	-14		
14.	-15		
15.	-16		

Sum: 0
Size: 0

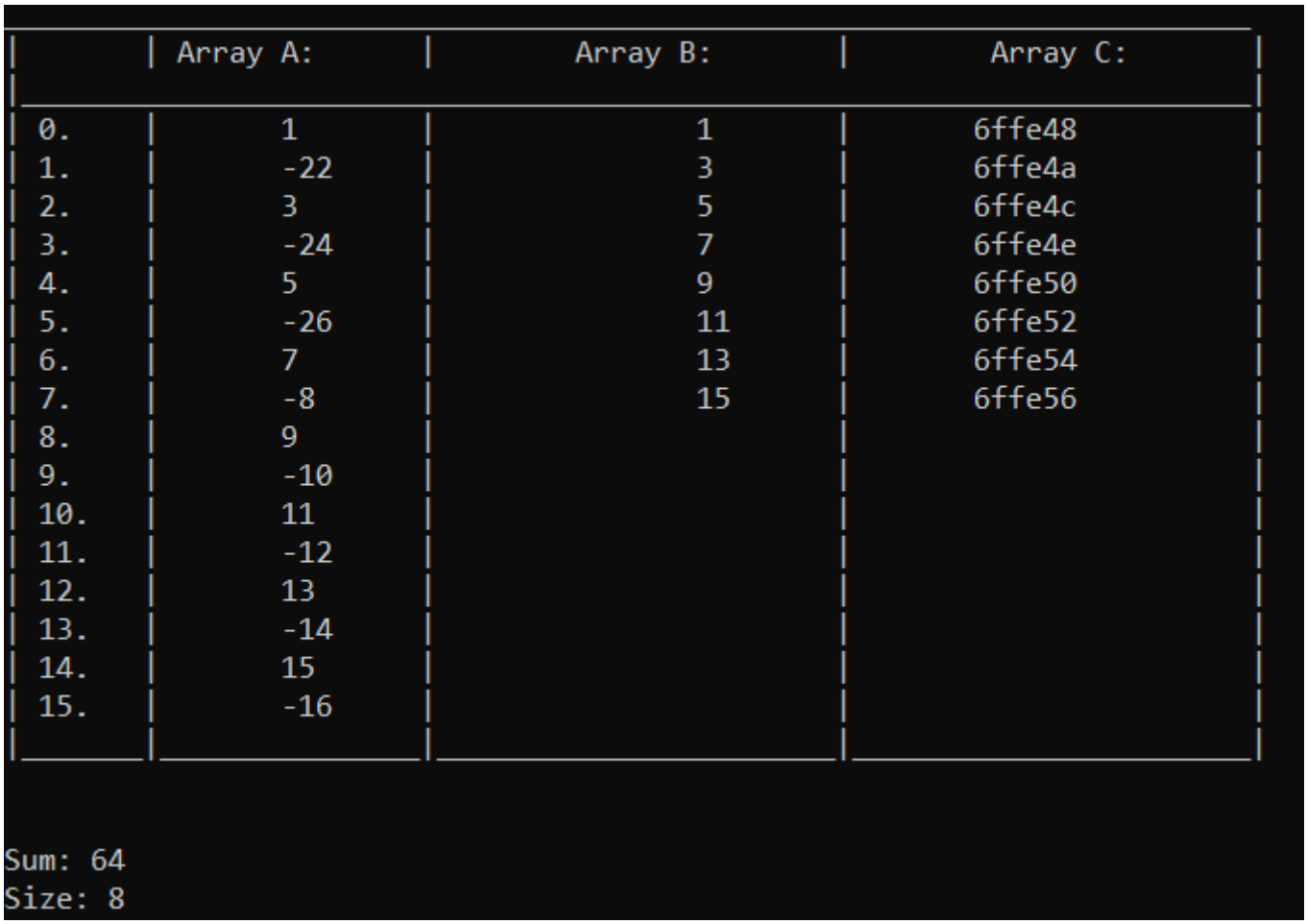
Все значения доступны для суммирования, сумма 0, количество допустимых значений – 16.



	Array A:	Array B:	Array C:
0.	0	0	12ff79c
1.	0	0	12ff79d
2.	0	0	12ff79e
3.	0	0	12ff79f
4.	0	0	12ff7a0
5.	0	0	12ff7a1
6.	0	0	12ff7a2
7.	0	0	12ff7a3
8.	0	0	12ff7a4
9.	0	0	12ff7a5
10.	0	0	12ff7a6
11.	0	0	12ff7a7
12.	0	0	12ff7a8
13.	0	0	12ff7a9
14.	0	0	12ff7aa
15.	0	0	12ff7ab

Sum: 0
Size: 16

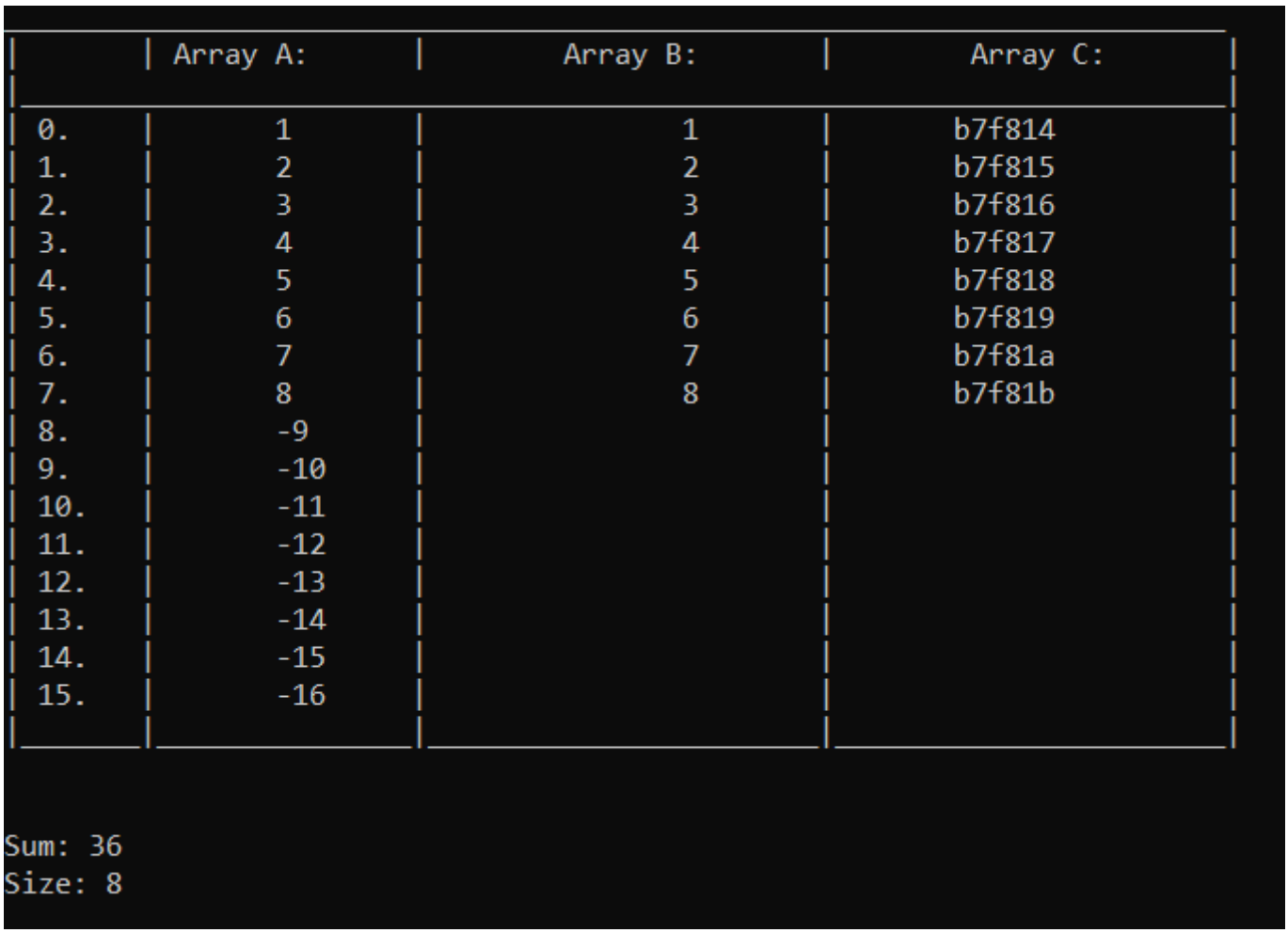
Не все значения доступны для суммирования, сумма 64, количество допустимых значений – 8.



	Array A:	Array B:	Array C:
0.	1	1	6ffe48
1.	-22	3	6ffe4a
2.	3	5	6ffe4c
3.	-24	7	6ffe4e
4.	5	9	6ffe50
5.	-26	11	6ffe52
6.	7	13	6ffe54
7.	-8	15	6ffe56
8.	9		
9.	-10		
10.	11		
11.	-12		
12.	13		
13.	-14		
14.	15		
15.	-16		

Sum: 64
Size: 8

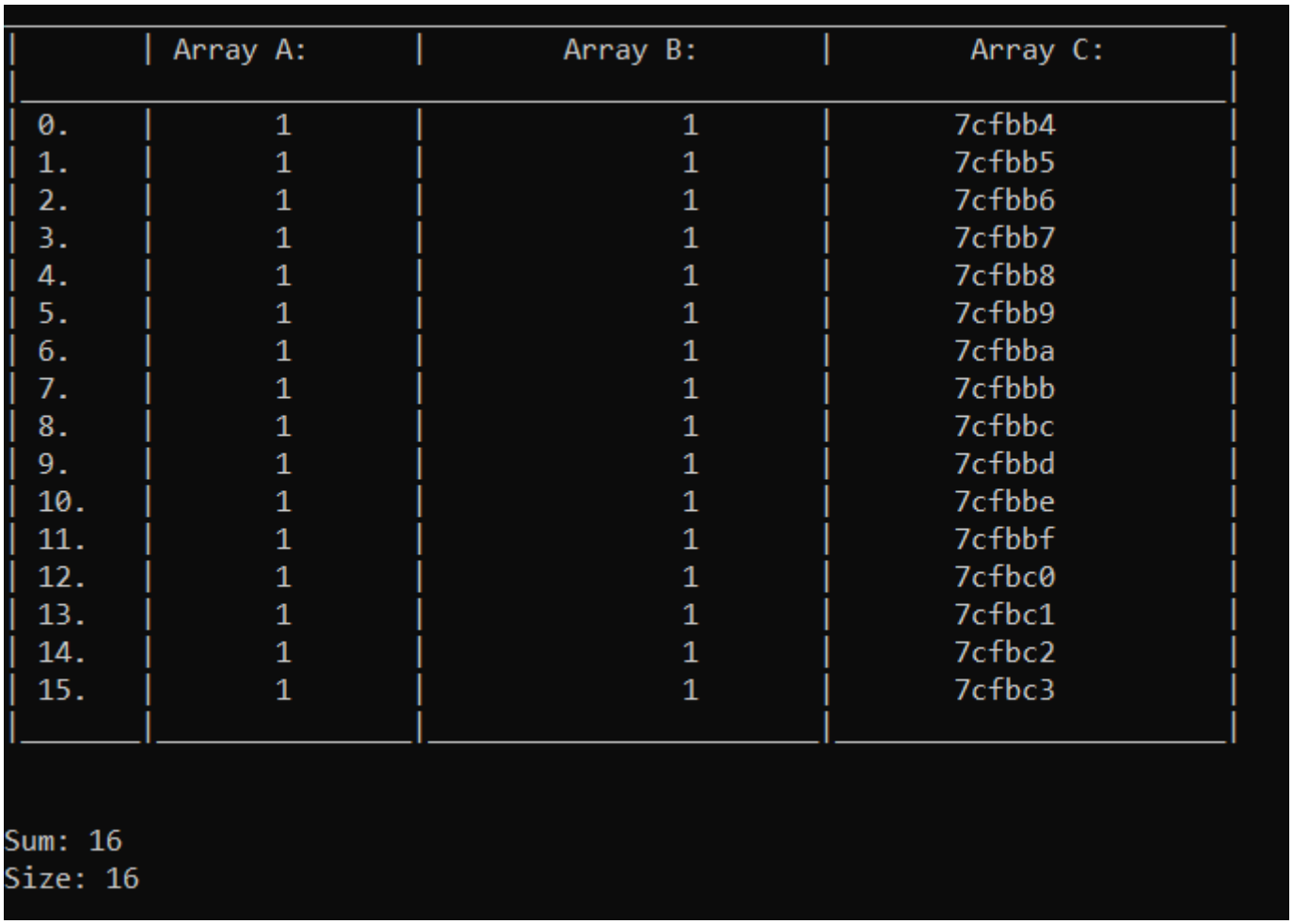
Доступны для суммирования только первые 8 элементов, сумма 36, количество допустимых значений – 8.



	Array A:	Array B:	Array C:
0.	1	1	b7f814
1.	2	2	b7f815
2.	3	3	b7f816
3.	4	4	b7f817
4.	5	5	b7f818
5.	6	6	b7f819
6.	7	7	b7f81a
7.	8	8	b7f81b
8.	-9		
9.	-10		
10.	-11		
11.	-12		
12.	-13		
13.	-14		
14.	-15		
15.	-16		

Sum: 36
Size: 8

Все значения доступны для суммирования, сумма 16, количество допустимых значений – 16.



	Array A:	Array B:	Array C:
0.	1	1	7cfbb4
1.	1	1	7cfbb5
2.	1	1	7cfbb6
3.	1	1	7cfbb7
4.	1	1	7cfbb8
5.	1	1	7cfbb9
6.	1	1	7cfbba
7.	1	1	7cfbbb
8.	1	1	7cfbbc
9.	1	1	7cfbbd
10.	1	1	7cfbbe
11.	1	1	7cfbbf
12.	1	1	7cfbc0
13.	1	1	7cfbc1
14.	1	1	7cfbc2
15.	1	1	7cfbc3

Sum: 16
Size: 16