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 №4.2
по курсу «Язык ассемблера»

Вариант №33

Ф.И.О. студента	Номер группы	Дата	Баллы
Николаев Александр Александрович	СКБ191	11.04.2022	

Задание А4

Москва, 2022 г.

Задача

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

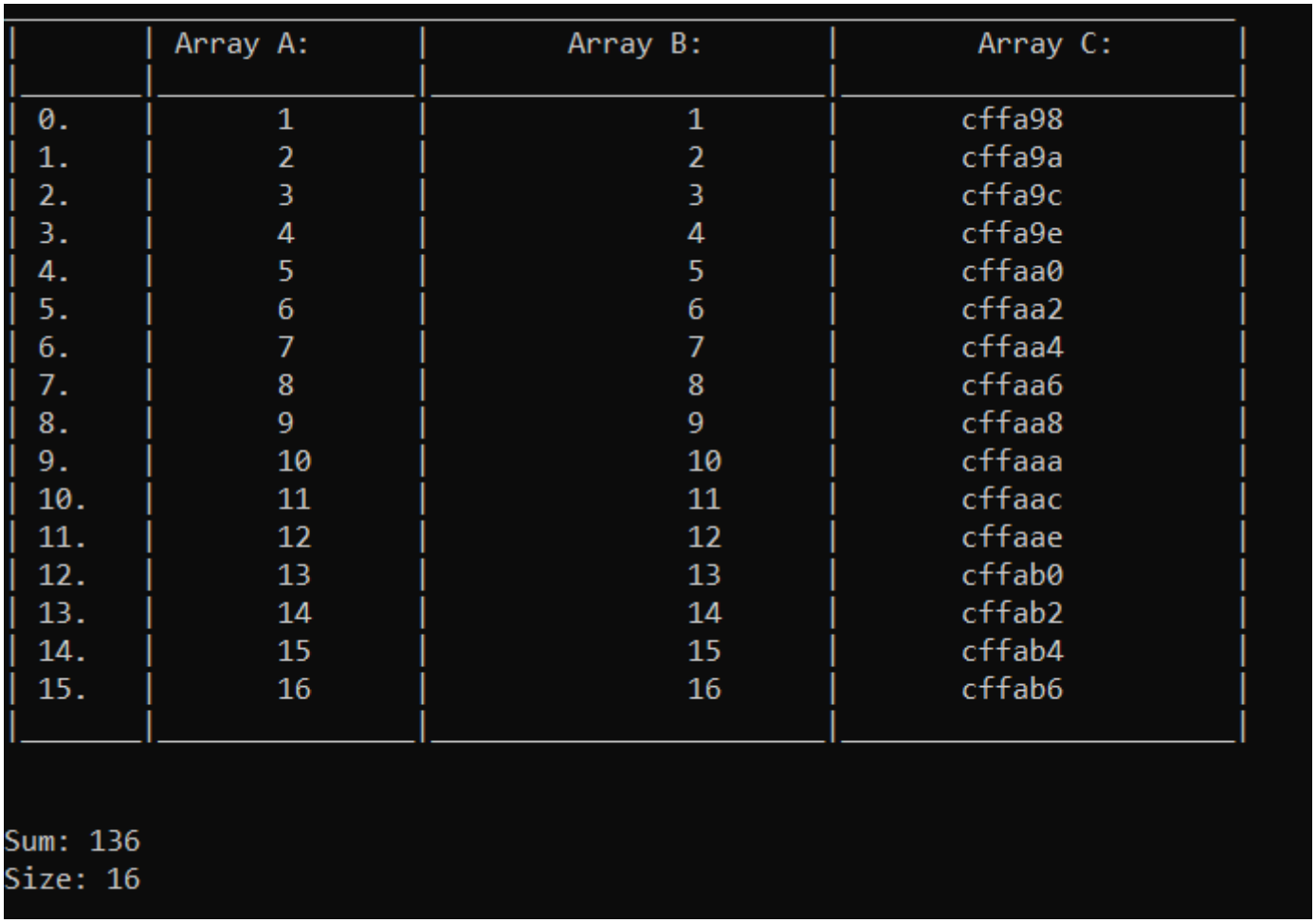
Текст программы

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

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

	int item;

	short int siz = 0;
	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) <= SHRT_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 eax, 0;			// под расширение чисел из массива A
		mov edx, 0;			// под сумму чисел из массива A

	BODY: cmp word 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 ax, [esi];          // скопировать значение элемента массива A в регистр
ax (как слово)

            mov [edi], ax;          // поместить в элемент массива B значение подходящего
элемента из массива A
            add edi, 2;             // 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

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

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

      NEXT: add esi, 2;             // 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, edx;           // 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	cffa98
1.	2	2	cffa9a
2.	3	3	cffa9c
3.	4	4	cffa9e
4.	5	5	cffaa0
5.	6	6	cffaa2
6.	7	7	cffaa4
7.	8	8	cffaa6
8.	9	9	cffaa8
9.	10	10	cffaaa
10.	11	11	cffaac
11.	12	12	cffaae
12.	13	13	cffab0
13.	14	14	cffab2
14.	15	15	cffab4
15.	16	16	cffab6

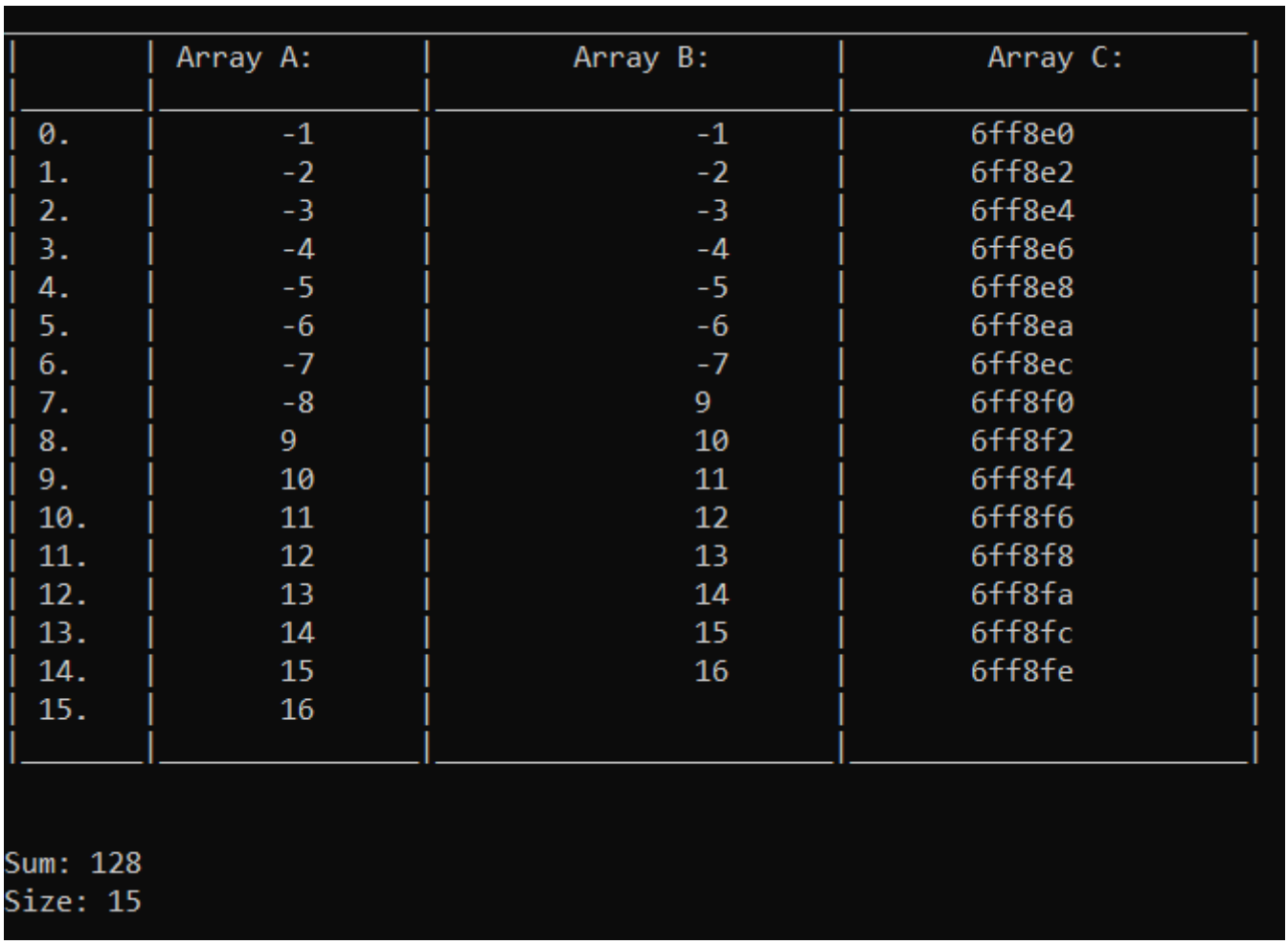
Sum: 136
Size: 16

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

	Array A:	Array B:	Array C:
0.	-1	-1	bdf730
1.	-2	-2	bdf732
2.	-3	-3	bdf734
3.	-4	-4	bdf736
4.	-5	-5	bdf738
5.	-6	-6	bdf73a
6.	-7	-7	bdf73c
7.	8	8	bdf73e
8.	9	9	bdf740
9.	10	10	bdf742
10.	11	11	bdf744
11.	12	12	bdf746
12.	13	13	bdf748
13.	14	14	bdf74a
14.	15	15	bdf74c
15.	16	16	bdf74e

Sum: 136
Size: 16

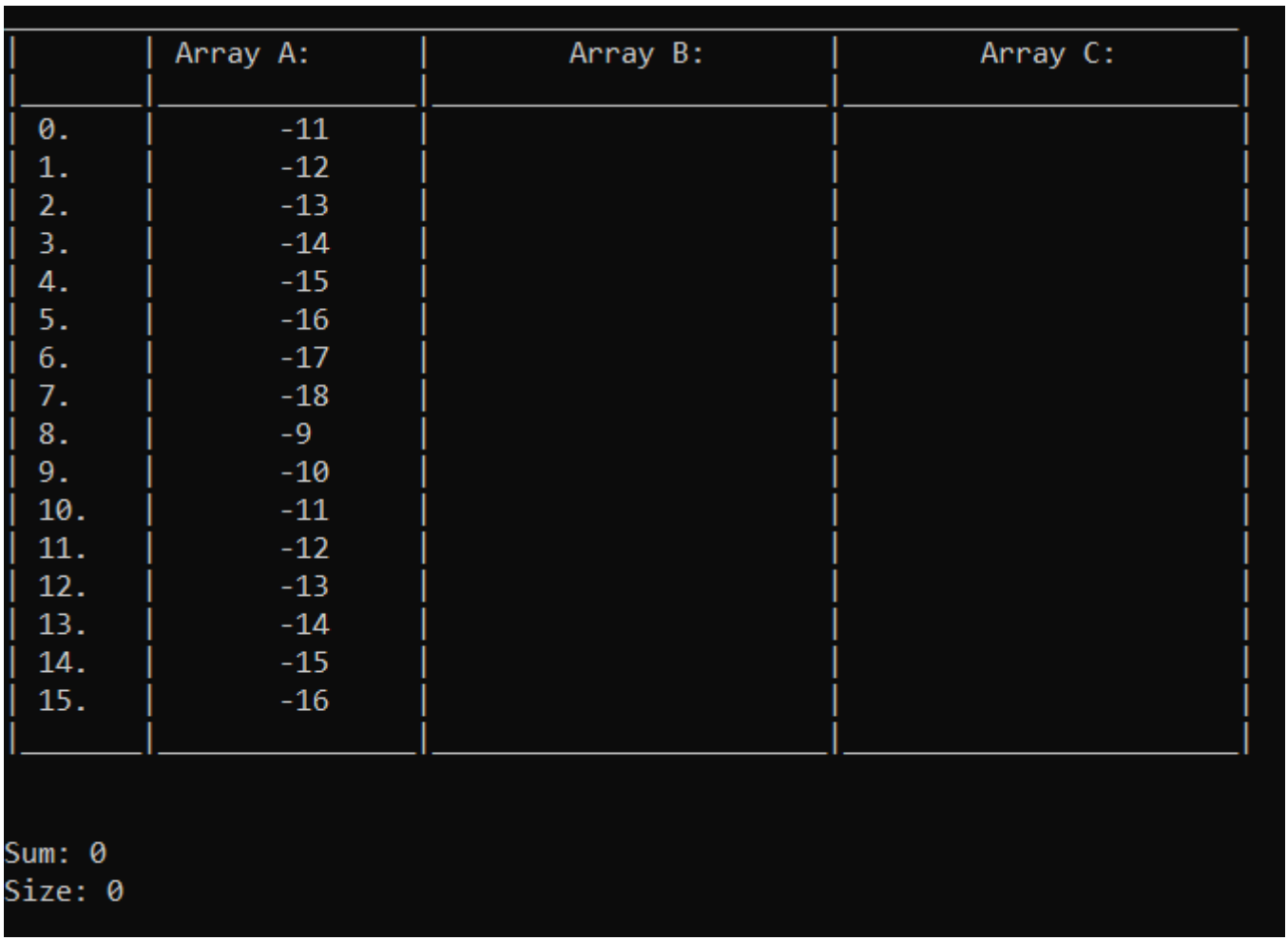
Доступны для суммирования все элементы, кроме -8, сумма 128, количество допустимых значений – 15.



	Array A:	Array B:	Array C:
0.	-1	-1	6ff8e0
1.	-2	-2	6ff8e2
2.	-3	-3	6ff8e4
3.	-4	-4	6ff8e6
4.	-5	-5	6ff8e8
5.	-6	-6	6ff8ea
6.	-7	-7	6ff8ec
7.	-8	9	6ff8f0
8.	9	10	6ff8f2
9.	10	11	6ff8f4
10.	11	12	6ff8f6
11.	12	13	6ff8f8
12.	13	14	6ff8fa
13.	14	15	6ff8fc
14.	15	16	6ff8fe
15.	16		

Sum: 128
Size: 15

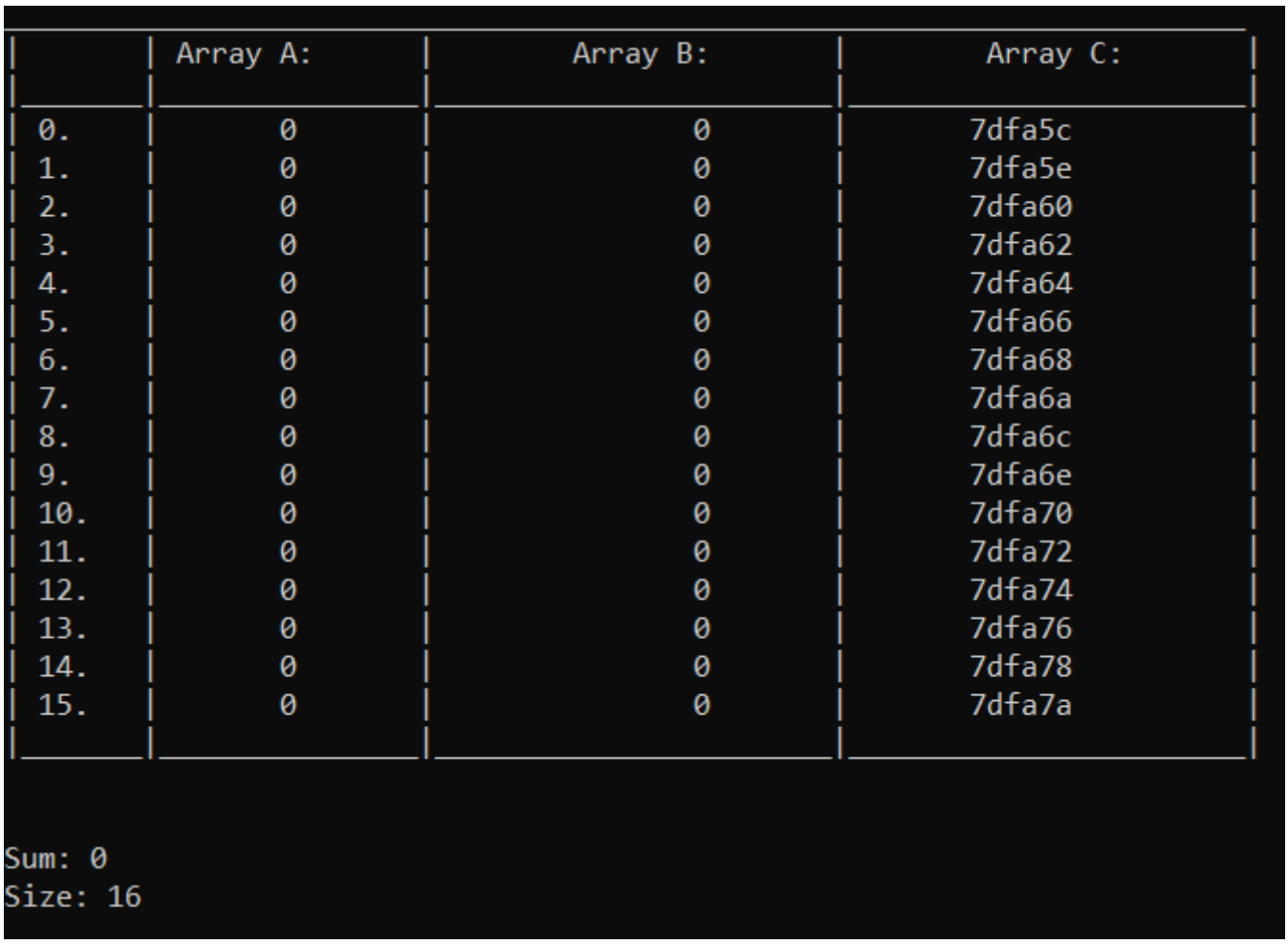
Ни одно значение недоступно для суммирования, сумма 0, количество допустимых значений – 0.



	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

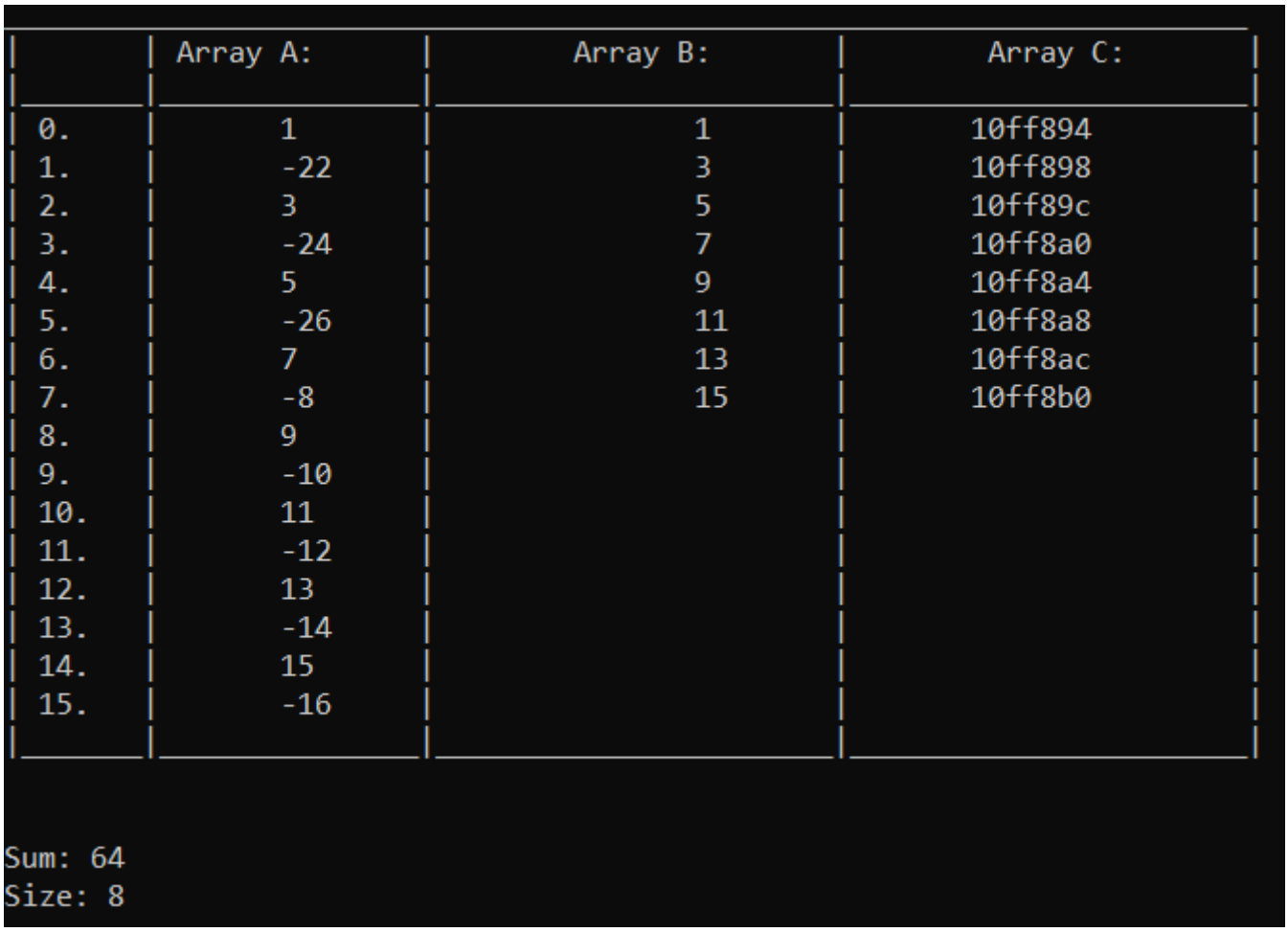
Все значения доступны для суммирования, сумма 0, количество допустимых значений – 16.



	Array A:	Array B:	Array C:
0.	0	0	7dfa5c
1.	0	0	7dfa5e
2.	0	0	7dfa60
3.	0	0	7dfa62
4.	0	0	7dfa64
5.	0	0	7dfa66
6.	0	0	7dfa68
7.	0	0	7dfa6a
8.	0	0	7dfa6c
9.	0	0	7dfa6e
10.	0	0	7dfa70
11.	0	0	7dfa72
12.	0	0	7dfa74
13.	0	0	7dfa76
14.	0	0	7dfa78
15.	0	0	7dfa7a

Sum: 0
Size: 16

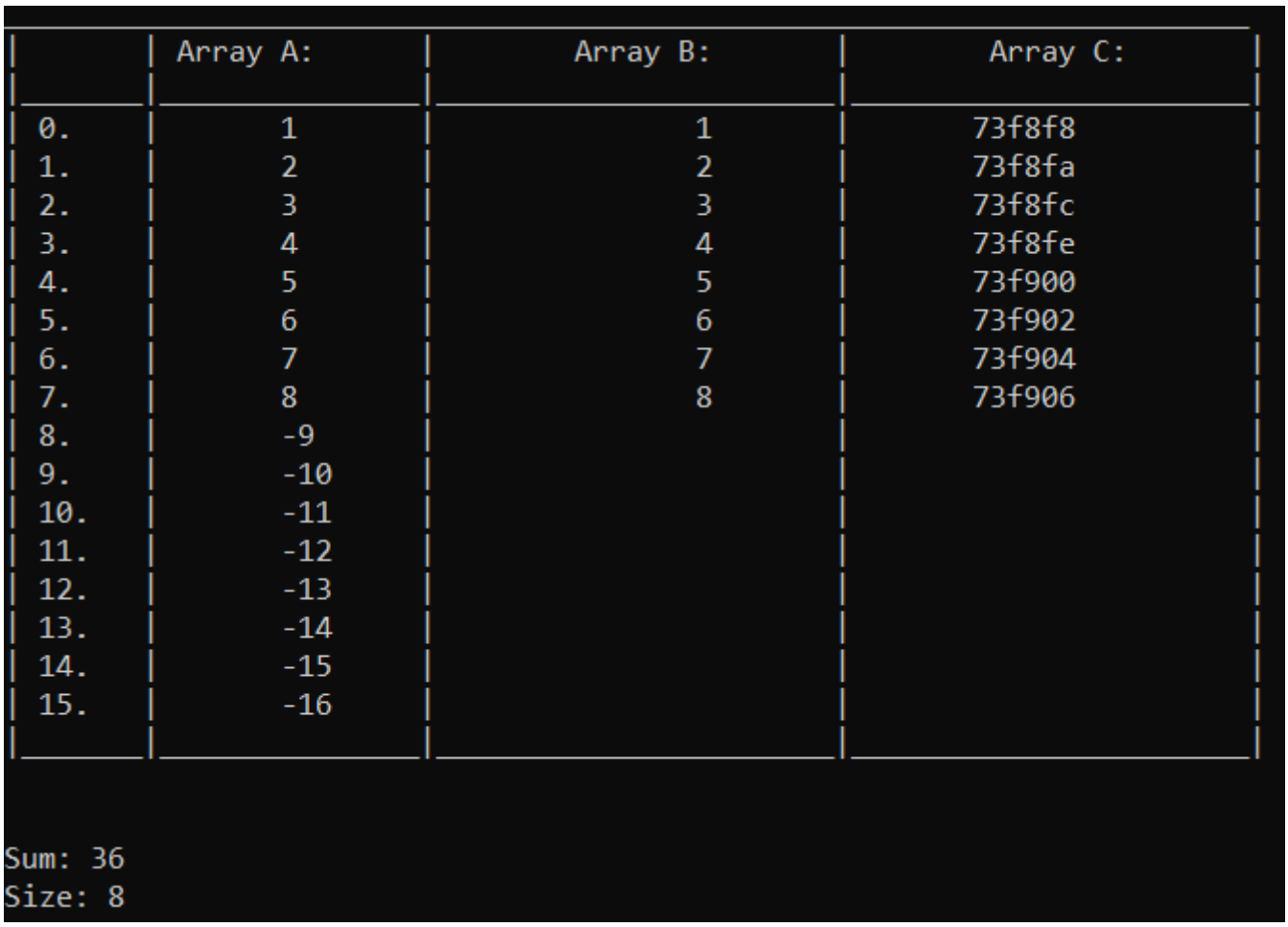
Не все значения доступны для суммирования, сумма 64, количество допустимых значений – 8.



	Array A:	Array B:	Array C:
0.	1	1	10ff894
1.	-22	3	10ff898
2.	3	5	10ff89c
3.	-24	7	10ff8a0
4.	5	9	10ff8a4
5.	-26	11	10ff8a8
6.	7	13	10ff8ac
7.	-8	15	10ff8b0
8.	9		
9.	-10		
10.	11		
11.	-12		
12.	13		
13.	-14		
14.	15		
15.	-16		

Sum: 64
Size: 8

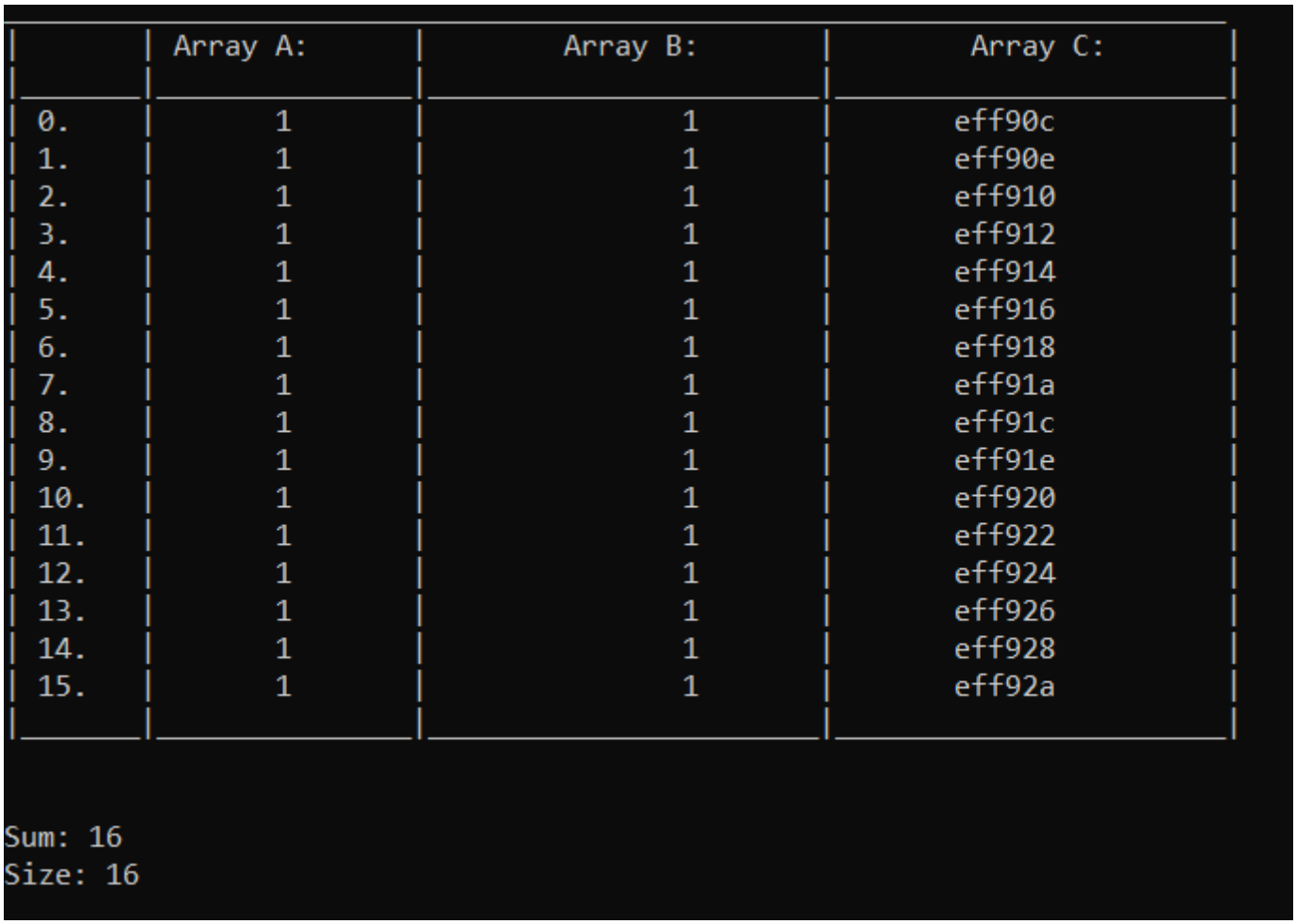
Доступны для суммирования только первые 8 элементов, сумма 36, количество допустимых значений – 8.



	Array A:	Array B:	Array C:
0.	1	1	73f8f8
1.	2	2	73f8fa
2.	3	3	73f8fc
3.	4	4	73f8fe
4.	5	5	73f900
5.	6	6	73f902
6.	7	7	73f904
7.	8	8	73f906
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	eff90c
1.	1	1	eff90e
2.	1	1	eff910
3.	1	1	eff912
4.	1	1	eff914
5.	1	1	eff916
6.	1	1	eff918
7.	1	1	eff91a
8.	1	1	eff91c
9.	1	1	eff91e
10.	1	1	eff920
11.	1	1	eff922
12.	1	1	eff924
13.	1	1	eff926
14.	1	1	eff928
15.	1	1	eff92a

Sum: 16
Size: 16