

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ»

“ВЫСШАЯ ШКОЛА ЭКОНОМИКИ”

Московский институт электроники и математики им. А.Н.Тихонова

ОТЧЁТ

По домашней работе 4.1

По курсу “Компьютерный практикум”

Вариант 81

ФИО студента	Группа	Дата	Балл
Волков Егор Ильич	БПМ245	23.05.2025	

Москва 2024

Задание А3:

81.

Даны два массива А и В (знаковые), состоящие из 16 байт каждый. Разделить элементы массивов А и В на пары. Если сумма элементов в соответствующих парах из А и В совпадают в байтовом диапазоне, то сосчитать количество таких пар и поместить адреса минимальных элементов таких пар из массивов А и В в массив С.

Ассемблерная вставка:

```
__asm
{
    lea esi, A; ESI = адрес A
    lea edi, B; EDI = адрес B
    lea ebx, C; EBX = адрес C
    mov ecx, N; счетчик циклов

    xor ax, ax; обнуляем AX для счетчика counter
    mov counter, ax

loop_start :
    cmp ecx, 0
    je loop_end

    movsx eax, byte ptr[esi]; eax = A[i], со знаком
    movsx edx, byte ptr[edi]; edx = B[i], со знаком

    add eax, edx; сумма пары

    cmp eax, 127
    jg skip_pair; если сумма > 127, пропускаем
    cmp eax, -128
    jl skip_pair; если сумма < -128, пропускаем

    ; Увеличиваем счетчик
    mov ax, counter
    inc ax
    mov counter, ax

    ; Записываем адреса минимальных элементов пары в C(по 2 адреса: A и B)
    movsx eax, byte ptr[esi]; eax = A[i]
    movsx edx, byte ptr[edi]; edx = B[i]

    cmp eax, edx
    jle a_min

    ; B[i] меньше, сначала адрес B[i]
    mov eax, edi
    mov[ebx], eax
    add ebx, 4

    mov eax, esi
    mov[ebx], eax
    add ebx, 4

    jmp next_iter

a_min :
    ; A[i] меньше или равен, сначала адрес A[i]
    mov eax, esi
    mov[ebx], eax
    add ebx, 4

    mov eax, edi
    mov[ebx], eax
    add ebx, 4

next_iter:
    inc esi
    inc edi
    dec ecx
    jmp loop_start

skip_pair :
    inc esi
    inc edi
    dec ecx
    jmp loop_start

loop_end :
    nop
}
```

Полный код:

```

#include <stdio.h>
#include <stdlib.h>
#define N 16

void CreateMassive(char* A)
{
    int input;
    for (int i = 0; i < N; i++)
    {
        scanf_s("%d", &input);
        A[i] = (char)input;
    }
}

void PrintMassive(char* A, int n)
{
    if (n == 1)
        printf("A[15] = {");
    else
        printf("\nB[15] = {");
    for (int i = 0; i < N; i++)
        printf("%d, ", A[i]);
    printf("\b");
}

int main()
{
    char A[16] = {
        1, 0, 90, 250,
        -5, 127, -120, 64,
        42, -73, 33, -1,
        100, -42, 0, 7
    };

    char B[16] = {
        2, 0, 70, 13,
        -10, 120, -120, 55,
        30, -90, 25, -2,
        99, -30, 1, 8
    };

    int C[2 * N] = { 0 }; // сюда пишем адреса минимальных элементов пар, по 2 адреса на пару (A и B)
    short counter = 0;

    PrintMassive(A, 1);
    PrintMassive(B, 2);

    __asm
    {
        lea esi, A; ESI = адрес A
        lea edi, B; EDI = адрес B
        lea ebx, C; EBX = адрес C
        mov ecx, N; счетчик циклов

        xor ax, ax; обнуляем AX для счетчика counter
        mov counter, ax

    loop_start :
        cmp ecx, 0
        je loop_end

        movsx eax, byte ptr[esi]; eax = A[i], со знаком
        movsx edx, byte ptr[edi]; edx = B[i], со знаком

        add eax, edx; сумма пары

        cmp eax, 127
        jg skip_pair; если сумма > 127, пропускаем
        cmp eax, -128
        jl skip_pair; если сумма < -128, пропускаем

        ; Увеличиваем счетчик
        mov ax, counter
        inc ax
        mov counter, ax
    }
}
```

```

; Увеличиваем счетчик
mov ax, counter
inc ax
mov counter, ax

; Записываем адреса минимальных элементов пары в C(по 2 адреса: A и B)
movsx eax, byte ptr[esi]; eax = A[i]
movsx edx, byte ptr[edi]; edx = B[i]

cmp eax, edx
jle a_min

; B[i] меньше, сначала адрес B[i]
mov eax, edi
mov[ebx], eax
add ebx, 4

mov eax, esi
mov[ebx], eax
add ebx, 4

jmp next_iter

a_min :
; A[i] меньше или равен, сначала адрес A[i]
mov eax, esi
mov[ebx], eax
add ebx, 4

mov eax, edi
mov[ebx], eax
add ebx, 4

next_iter:
inc esi
inc edi
dec ecx
jmp loop_start

skip_pair :
inc esi
inc edi
dec ecx
jmp loop_start

loop_end :
nop
}

printf("\nКоличество подходящих пар: %d\n", counter);
printf("Адреса минимальных элементов пар и их значения:\n");

for (int i = 0; i < counter * 2; i += 2)
{
    char* ptr1 = (char*)C[i];
    char* ptr2 = (char*)C[i + 1];
    printf("Пара %d: min addr = %p, value = %d\n",
        i / 2 + 1, ptr1, *ptr1);
}

return 0;
}

```

Тесты:

```

A[15] = {1, 0, 90, 2, -5, 127, -128, 64, 42, -73, 33, -1, 100, -42, 0, 7,}
B[15] = {2, 0, 70, 13, -10, 120, -120, 55, 30, -90, 25, -2, 99, -30, 1, 8,}
Количество подходящих пар: 11
Адреса минимальных элементов пар и их значения:
Пара 1: min addr = 0137FB90, value = 1
Пара 2: min addr = 0137FB91, value = 0
Пара 3: min addr = 0137FB93, value = 2
Пара 4: min addr = 0137FB7C, value = -10
Пара 5: min addr = 0137FB7F, value = 55
Пара 6: min addr = 0137FB80, value = 30
Пара 7: min addr = 0137FB82, value = 25
Пара 8: min addr = 0137FB83, value = -2
Пара 9: min addr = 0137FB9D, value = -42
Пара 10: min addr = 0137FB9E, value = 0
Пара 11: min addr = 0137FB9F, value = 7

```

```
A[15] = {3, 0, 0, 12, -5, -89, 52, 64, 42, -33, 36, -1, -56, -12, 0, 0,}
B[15] = {2, 0, 70, 13, -10, 120, -120, 55, 30, -90, 25, -2, 99, -30, 1, 8,}
Количество подходящих пар: 16
Адреса минимальных элементов пар и их значения:
Пара 1: min addr = 00F3FDC4, value = 2
Пара 2: min addr = 00F3FDDD, value = 0
Пара 3: min addr = 00F3FDDE, value = 0
Пара 4: min addr = 00F3FDDF, value = 12
Пара 5: min addr = 00F3FDC8, value = -10
Пара 6: min addr = 00F3FDE1, value = -89
Пара 7: min addr = 00F3FDCA, value = -120
Пара 8: min addr = 00F3FDCB, value = 55
Пара 9: min addr = 00F3FDCC, value = 30
Пара 10: min addr = 00F3FDCE, value = -90
Пара 11: min addr = 00F3FDCE, value = 25
Пара 12: min addr = 00F3FDCE, value = -2
Пара 13: min addr = 00F3FDE8, value = -56
Пара 14: min addr = 00F3FDD1, value = -30
Пара 15: min addr = 00F3FDEA, value = 0
Пара 16: min addr = 00F3FDEB, value = 0
```