**Правительство Российской Федерации**

**Федеральное государственное автономное образовательное учреждение высшего профессионального образования   
"Национальный исследовательский университет   
"Высшая школа экономики"**

Московский институт электроники и математики Национального

исследовательского университета "Высшая школа экономики"

Департамент прикладной математики

**ОТЧЕТ**

**По лабораторной работе №А4.1**

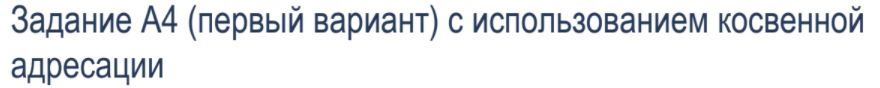
**По курсу «Компьютерный практикум»**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| |  |  | | --- | --- | |  | ФИО студента | | Номер группы | Дата |
| Кармаев Александр Андреевич | БПМ211 | 13.05.2022 |
|  |
|  |
|  |

**Москва – 2022 г.**

1. **ЗАДАНИЕ А4.1**

Даны массивы A и B из 16 **слов**. Если элемент массива A меньше соответствующего элемента массива B (числа беззнаковые), то поменяйте местами эти элементы. Подсчитайте количество обменов. В массив С поместите адреса (смещения) этих элементов.



Изображение выглядит как текст

Автоматически созданное описание

1. **РЕШЕНИЕ**

#include <stdio.h>

#define N 16 // размер массивов

int main() {

int amount = 0; // число обменов

short int A[N]; // массивы элементов

short int B[N];

short int \*C[N \* 2]; // массив адресов смещения элементов

// вводим элементы в массивы

printf(" Input data in array A\n");

for (int i = 0; i < N; i++) {

printf("A[%d] = ", i);

scanf("%d", &A[i]);

}

printf("\n Input data in array B\n");

for (int i = 0; i < N; i++) {

printf("B[%d] = ", i);

scanf("%d", &B[i]);

}

// показываем содержимое массивов до обработки

printf("\n Array A befor processing: ");

for (int i = 0; i < N; i++) {

printf("%d ", A[i]);

}

printf("\n Array B befor processing: ");

for (int i = 0; i < N; i++) {

printf("%d ", B[i]);

}

\_\_asm {

// установка начальных значений

sub ecx, ecx // счётчик цикла и адресация в массивах A и B

mov ebx, ecx // адресация в массиве С и подсчёт обменов

// начало цикла

step: cmp ecx, LENGTH A // массивы обработаны

jae end // если ДА - массивы обработаны

// иначе сравниваем значения в одинаковых позициях массивом

mov dx, A[ecx \* TYPE A] // помещаем элемент массива А в регистр

cmp dx, B[ecx \* TYPE B] // сравнииеваем

jge no // если значения флага соответствует отношению A[i] >= B[i], то переходим к метке(no), иначе условие выполнено

lea eax, A[ecx \* TYPE A] // помещаем в eax адрес текущего элемента из А

mov C[ebx \* TYPE C], eax // помещаем в массив С адрес смещения

inc ebx // переходим к следующему элементу массива С

lea eax, B[ecx \* TYPE B] // помещаем в eax адрес текущего элемента из B

mov C[ebx \* TYPE C], eax // помещаем в массив С адрес смещения

inc ebx // переходим к следующему элементу массива С

xchg dx, B[ecx \* TYPE B] // меняем местами элементы A и B массивов

mov A[ecx \* TYPE A], dx

// если уловие не выполнено, то есть A[i] >= B[i]

no:

inc ecx // переходим к следующему элементу массивов А и В

jmp step // идём в начало цикла

// обработали все элементы массивов А и В

end:

mov amount, ebx // помещаем в переменную количество смещений, но их в два раза больше, чем обменов

}

// показываем содержимое массивов после обработки

printf("\n\n Array A after processing: ");

for (int i = 0; i < N; i++) {

printf("%d ", A[i]);

}

printf("\n Array B after processing: ");

for (int i = 0; i < N; i++) {

printf("%d ", B[i]);

}

// показываем количество изменений и адреса смещений

printf("\n\n There is %d changes: \n", amount / 2);

for (int i = 0; i < amount / 2; i++) {

printf("A[%d] = %p ", i, C[2 \* i]);

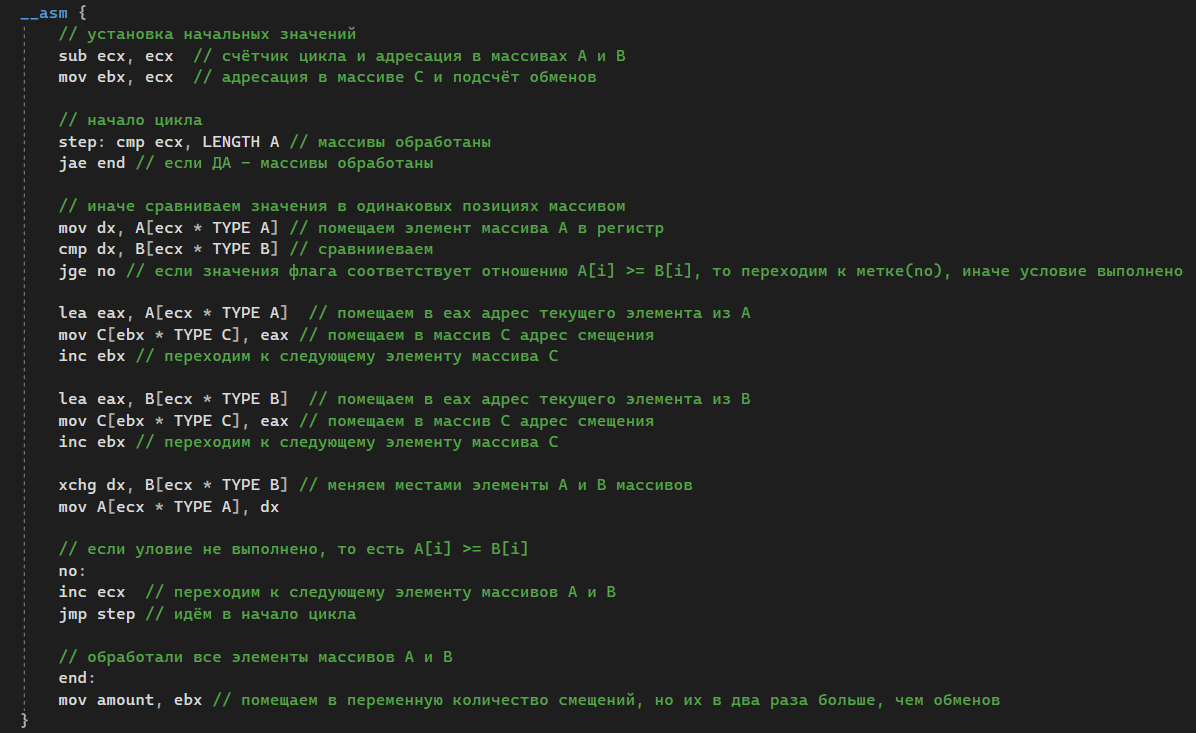
printf("B[%d] = %p\n", i, C[2 \* i + 1]);

}

exit(0);

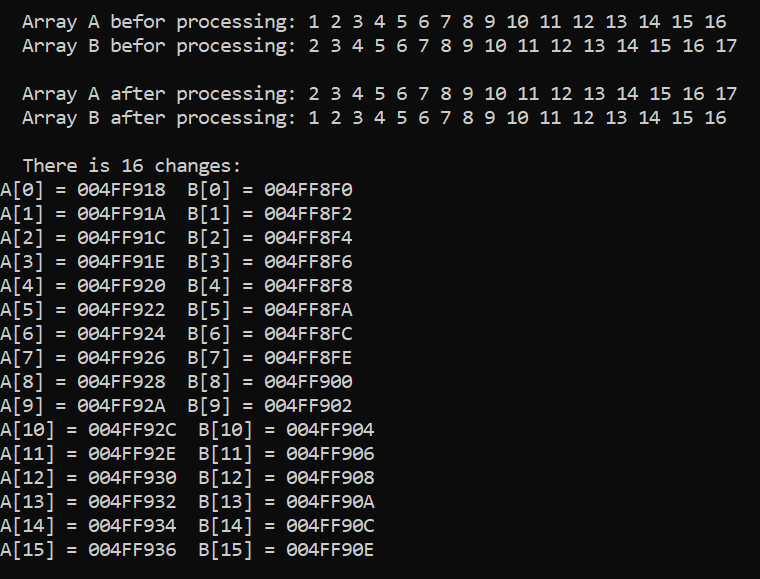
}

**Ассемблерная вставка:**

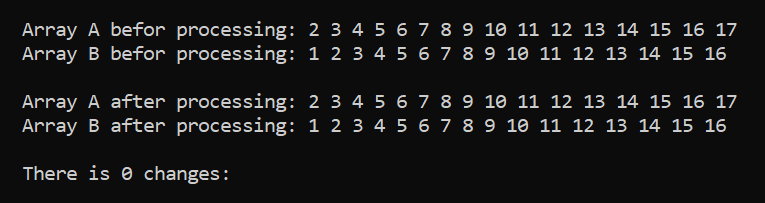


1. **Тесты**

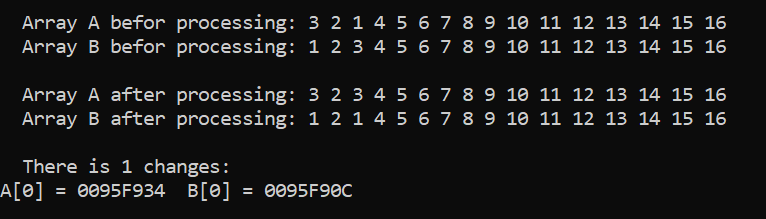
// меняем все элементы

****

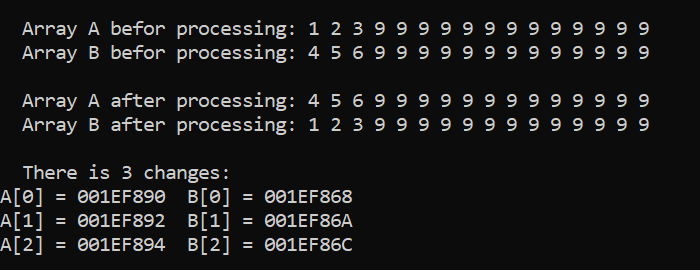
// элементы не меняются

****

// одно изменение

****

// несколько изменений

****