МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования
Санкт-Петербургский национальный исследовательский университет ИТМО
Мегафакультет трансляционных информационных технологий

Факультет информационных технологий и программирования

Дополнительная работа №1. Сортировка

По дисциплине «Аппаратное обеспечение вычислительных систем»

Вариант №

Выполнил студент группы
№M3112
Тимофеев Вячеслав [redacted]

Проверила
Шевчик [redacted]

УНИВЕРСИТЕТ ИТМО

Санкт-Петербург
2024

Условие

Создать программу на языке C, которая считывает из файла целые числа.

Необходимо написать ассемблерную вставку, **осуществляющую сортировку массива**. Метод сортировки определяет разработчик. Вставка должна быть вынесена в отдельную функцию. Использовать глобальные переменные для передачи данных в указанную функцию **запрещено**.

Полученный массив записать в файл вывода.

Входные данные

Первая строка входного файла INPUT.TXT содержит одно целое число N (1 ≤ N ≤ 100).

В следующих N строках содержатся числа, по модулю не превышающие 10^9.

Код:

main.cpp × INPUT.txt OUTPUT.txt

```cpp
#include <stdio.h>
#define F FILE
typedef long long ll;

void sortContainer(ll *Container, ll n) {
    asm(
            "mov %[Container_ptr], %%rsi;"  // rsi = Container_ptr
            "mov %[N], %%rcx;"          //  rcx=N

            "out:;"
            "mov $1, %%rax;"            // j=1

            "next:;"
            "cmp %%rcx, %%rax;"         // rcx=rax ? end : do
            "jge end;"

            "mov (%%rsi, %%rax, 8), %%rbx;"  // Загружаем текущий элемент в rbx
            "mov %%rax, %%rdx;"          // rax=rdx
            "dec %%rdx;"                 // rdx--

            "in:;"
            "cmp $0, %%rdx;"             // rdx=0 ? go swap : do
            "jl insert;"
            "mov (%%rsi, %%rdx, 8), %%rdi;"

            "cmp %%rdi, %%rbx;"          // rdi>=rbx ? insert : do
            "jge insert;"

            "mov %%rdi, 8(%%rsi, %%rdx, 8);"  // Сдвигаем предыдущий элемент вправо
            "dec %%rdx;" // j--
            "jmp in;"   // next i

            "insert:;"
            "mov %%rbx, 8(%%rsi, %%rdx, 8);"  // Вставляем текущий элемент на нужное место
            "inc %%rax;"                      // rax++
            "jmp next;"                       // for(i++)

            "end:;"
            :                                                 // Выходные операнды
            : [Container_ptr] "r" (Container), [N] "r" (n) // Входные операнды
    : "rax", "rbx", "rcx", "rdx", "rsi", "rdi"              // Используемые регистры
    );
}
```

```cpp
// rsi - i текущего элемента (Исходный индексный регистр)
// rax - для выполнения арифметических и логических операций (Накопительный регистр)
// rcx - счетчик цикла в операциях с повторяющимися командами и циклами (Регистр счетчика)
// rdi - указатель на данные, которые будут обрабатываться, особенно в строковых операциях (Целевой индексный регистр)
// rbx - для хранения данных и адресов (Базовый регистр)
// rdx - для хранения данных и в качестве вспомогательного регистра для некоторых операций (Регистр данных)
// ecx - текущий элемент массива (Расширенный регистр счетчика)
// edx - предыдущий элемент массива (Расширенный реестр данных)
// eax - для хранения арифм и лог операций (Накопительный регистр)

int main() {
    ll n;
    F *input = fopen( Filename: "input.txt",  Mode: "r");
    fscanf( stream: input,  format: "%lld", &n);
    ll Container[n];

    for (ll i = 0; i < n; i++) fscanf( stream: input,  format: "%lld", &Container[i]);
    fclose( File: input);

    sortContainer(Container, n);

    F *output = fopen( Filename: "output.txt",  Mode: "w");
    for (ll i = 0; i < n; i++) fprintf( stream: output,  format: "%lld ", Container[i]);

    fclose( File: output);
    return 0;
}
```

Пример: