МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования” Гомельский государственный университет им Ф. Скорины”

Отчет по лабораторной работе:

**КФ-Программирование-1-1-05 ИЗУЧЕНИЕ И ПРОГРАММИРОВАНИЕ АЛГОРИТМОВ СОРТИРОВКИ. ПРИМЕНЕНИЕ ГСЧ**

Выполнил:

Студент группы КФ-17

Гуревич П.А.

Проверил Дей Е.А.

Гомель 2022

**Цель работы:** изучение основных алгоритмов внутренней сортировки данных и их программная реализация

**ЗАДАЧА 1**. Разработать программу, в которой реализованы требуемые вычисления с необходимыми пояснениями:

1) Вывод на экран содержания задания

2) Вывод на экран и данных исполнителя

3) заполнение массива, содержащего заданное количество элементов типа integer, случайными числами в указанном диапазоне указанным методом (см. Таблицу вариантов);

4) вывод элементов заполненного массива

5) сортировка массива заданным методом с выводом элементов массива после каждого шага сортировки;

6) вывод результата сортировки;

**ЗАДАЧА 1**.

Разработать программу, в которой реализованы требуемые вычисления с необходимыми пояснениями:

1) Вывод на экран содержания задания

2) Вывод на экран и данных исполнителя

3) заполнение массива, содержащего заданное количество элементов типа integer, случайными числами в указанном диапазоне указанным методом (см. Таблицу вариантов);

4) вывод элементов заполненного массива

5) сортировка массива заданным методом с выводом элементов массива после каждого шага сортировки;

6) вывод результата сортировки;

using System;

Console.WriteLine("Метод четных и нечетных транспозиций");

Console.WriteLine("Выполнил Гуревич Павел Андреевич");

int[] a = new int[14];

Random rnd = new Random();

int per, i, temp;

for (i = 0; i < a.Length; i++)

a[i] = rnd.Next(12, 40);

Console.WriteLine("изначально");

for (i = 0; i < a.Length; i++)

Console.Write(" " + a[i]);

Console.WriteLine();

int k = 1;

do

{

per = 0;

i = 0;

while (i < a.Length - 1)

{

if (a[i] > a[i + 1])

{

temp = a[i];

a[i] = a[i + 1];

a[i + 1] = temp;

per = 1;

Console.WriteLine(" Итерация #" + k);

for (i = 0; i < a.Length; i++)

Console.Write(" " + a[i]);

k++;

Console.WriteLine();

}

i= i + 2;

}

i = 1;

while (i < a.Length - 1)

{

if (a[i] > a[i + 1])

{

temp = a[i];

a[i] = a[i + 1];

a[i + 1] = temp;

per = 1;

Console.WriteLine(" Итерация #" + k);

for (i = 0; i < a.Length; i++)

Console.Write(" " + a[i]);

k++;

Console.WriteLine();

}

i = i + 2;

}

} while (per != 0);

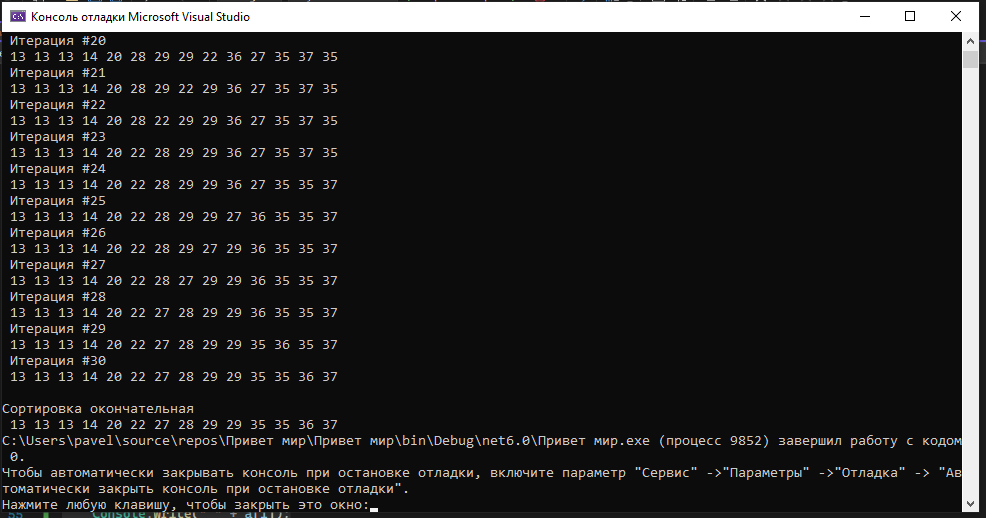
Console.WriteLine();

Console.WriteLine("Cортировка окончательная");

for (i = 0; i < a.Length; i++)

Console.Write(" " + a[i]);

Результат сортировки:



**КОНТРОЛЬНЫЕ ВОПРОСЫ**

1. Сортировка данных. Основные понятия:

Существует множество различных методов сортировки, причем для разных типов задач подходящими являются разные методы. Практически каждый метод сортировки содержит такие этапы: - сравнение, определяющее упорядоченность пары элементов; - перестановку, меняющую местами пару элементов; - способ организации операций сравнения и перестановки до тех пор, пока все элементы не будут упорядочены.

1. Описать алгоритм своего варианта метода сортировки

Этот метод является разновидностью метода обмена. Сортировка выполняется следующим образом. Массив просматривается многократно, причем на первом просмотре сравниваются A[i] с A[i+1] для всех нечетных i. На втором просмотре сравниваются A[i] с A[i+1] для всех четных i. Каждый раз, когда A[i]>A[i+1], выполняется обмен этих элементов. Просмотры продолжаются до тех пор, пока массив не будет упорядочен.

1. Записать текст процедуры, реализующей свой вариант метода сортировки

do

{

per = 0;

i = 0;

while (i < a.Length - 1)

{

if (a[i] > a[i + 1])

{

temp = a[i];

a[i] = a[i + 1];

a[i + 1] = temp;

per = 1;

}

i = i + 2;

}

i = 1;

while (i < a.Length - 1)

{

if (a[i] > a[i + 1])

{

temp = a[i];

a[i] = a[i + 1];

a[i + 1] = temp;

per = 1;

}

i = i + 2;

}

} while (per != 0);

1. Описать алгоритм следующего варианта метода сортировки

Последовательными сравнениями ищется наименьший элемент массива, причем в процессе поиска запоминается индекс текущего наименьшего числа. После того как просмотрен последний элемент, можно поменять местами 1- й элемент с наименьшим, т.к. его индекс известен. Затем эти действия повторяются, начиная со 2-го элемента (1-й уже стоит на правильном месте), затем с 3-го элемента, и т.д.

1. Записать текст процедуры, реализующей следующий вариант метода сортировки.

for (int i = 0; i < b.Length-2; i++)

{

min = b[i];

nom=i;

for(int j=i+1;j<b.Length-1; j++)

{

if (b[j] < min)

{

min = b[j];

nom = j;

}

}

K = b[i];

b[i] = b[nom];

b[nom] = K;

}

**Вывод:** изучил основных алгоритмов внутренней сортировки данных и их программную реализацию.