Лабораторная работа 1.

//№1.

#define \_USE\_MATH\_DEFINES

#include <iostream>

#include <math.h>

#include <cmath>

using namespace std;

int main()

{

setlocale(LC\_ALL, "Rus");

//Хасанов Алмаз 4333

double Y, gam, c, b, x, z, r, f;

cout << "Gamma = "; cin >> gam;

cout << "c = "; cin >> c;

cout << "b = "; cin >> b;

cout << "x = "; cin >> x;

cout << "z = "; cin >> z;

cout << "r = "; cin >> r;

cout << "f = "; cin >> f;

double a = b\*b;

a += c;

a = log(a);

double d = sin(x);

a /= d;

d = -(gam) + f;

d = pow(M\_E, d);

a += d;

d = pow(r, b);

a += d;

d = z + r;

d = sqrt(d);

Y = a + d;

cout << "Результат Y = " << Y <<endl;

system("pause");

return 0;

}

/////////////////////////////////////////////////////////////////\\

//№2

#include <iostream>

using namespace std;

int main()

{

setlocale(LC\_ALL, "Rus");

//Хасанов Алмаз 4333

int x, y, k;

cout << "X = "; cin >> x;

cout << "Y = "; cin >> y;

if (x >= 0)

if (y >= 0)

k = 1;

else

k = 4;

else

if (y >= 0)

k = 2;

else

k = 3;

if (x == 0)

if (y == 0)

cout << "Это начало координат" << endl;

else

cout << k << " четверть";

else

cout << k << " четверть" << endl;

system("pause");

return 0;

}

/////////////////////////////////////////////////////////////////\\

//№3

#define \_USE\_MATH\_DEFINES

#include <iostream>

#include <math.h>

#include <cmath>

using namespace std;

int main()

{

setlocale(LC\_ALL, "Rus");

//Хасанов Алмаз 4333

double a, b, n;

cout << "a = "; cin >> a;

cout << "b = "; cin >> b;

cout << "n = "; cin >> n;

double s = a + b;

double f = n++;

s /= f;

f = a;

f += s;

while (f < b)

{

int y = -0.5 \* f;

y = pow(M\_E, y);

y = 1 - y;

y \*= 5;

double c = pow(2, M\_PI);

c \*= f;

c = cos(c);

y \*= c;

cout << "Y = " << y << endl;

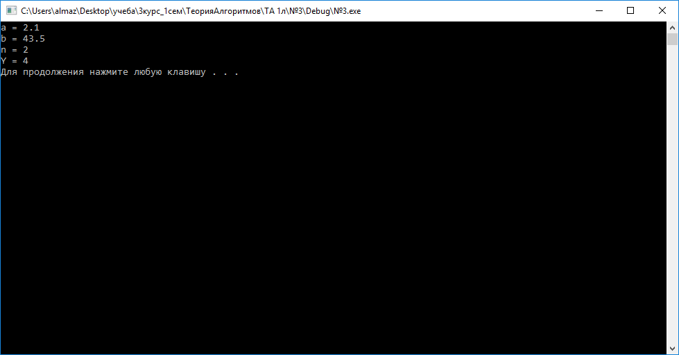
f += s;

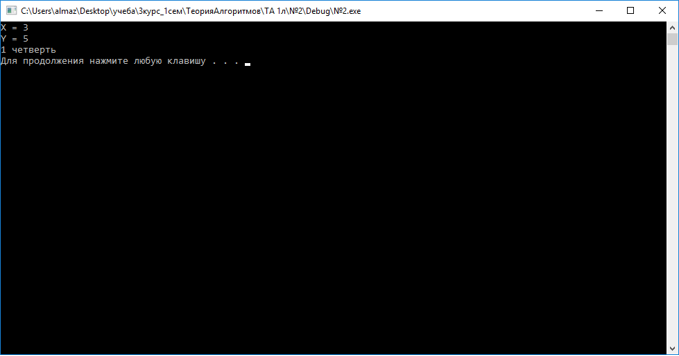
}

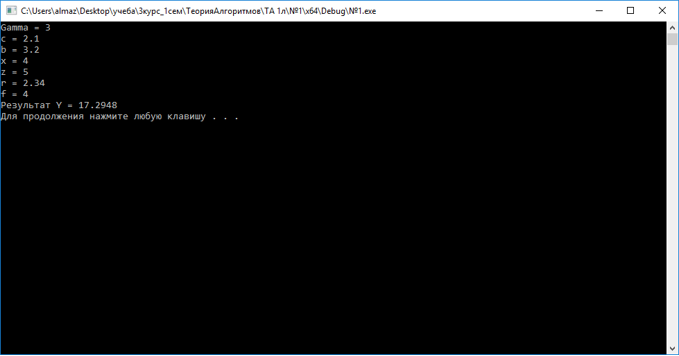
system("pause");

return 0;

}







Лабораторная работа 2.

#include <iostream>

//Хасанов Алмаз 4333

using namespace std;

void Input(int n, int \*mas)

{

cout << endl << "Введите массив: " << endl;

for (int i = 0; i < n; i++)

cin >> mas[i];

}

void Output(int n, int \*mas)

{

cout << endl;

for (int i = 0; i < n; i++)

cout << mas[i] << " ";

cout << endl;

}

void LineSearch(int n, int \*mas)

{

cout << "Введите значение для поиска:" << endl;

int value, pos = -1, i = 0;

cin >> value;

while ((pos == -1) && (i < n))

{

if (mas[i] == value)

pos = i;

i++;

}

if (pos == -1)

cout << "Значение не найдено!";

else

cout << "Позиция числа : " << pos + 1;

cout << endl;

}

int BinarSearch(int value, int n, int \*mas)

{

int left = 0, right = n - 1, midle;

while (left < right)

{

midle = ((left + right) / 2);

if (mas[midle] >= value)

right = midle;

else left = midle + 1;

}

if (value == mas[left])

return left;

}

void SelectSort(int n, int \*mas)

{

for (int i = 0; i<n - 1; i++)

for (int j = i; j < n; j++)

{

if (mas[i] > mas[j])

{

int a = mas[i];

mas[i] = mas[j];

mas[j] = a;

}

}

}

void BubbleSort(int n, int \*mas)

{

for (int i = 1; i < n; i++)

for (int j = i; j >= 1; j = j - 1)

if (mas[j - 1] > mas[j])

{

int a = mas[j - 1];

mas[j - 1] = mas[j];

mas[j] = a;

}

}

void IncludeSort(int n, int \*mas)

{

int a, pos;

for (int j = 0; j < n; j++)

{

a = mas[j];

pos = BinarSearch(a, n, mas);

while (pos < j)

{

for (int i = j - 1; i >= pos; i--)

mas[i + 1] = mas[i];

mas[pos] = a;

break;

}

}

}

void QuickSort(int \*mas, int left, int right)

{

int i, j, m;

i = left;

j = right;

m = mas[(i + j + 1) / 2];

while (i <= j)

{

while (mas[i] < m)

i++;

while (mas[j] > m)

j--;

if (i <= j)

{

int a = mas[i];

mas[i] = mas[j];

mas[j] = a;

i++;

j--;

}

}

if (left < j)

QuickSort(mas, left, j);

if (i < right)

QuickSort(mas, i, right);

}

void main()

{

int h, \*mas = new int[1], n;

setlocale(LC\_ALL, "Russian");

do

{

system("cls");

cout << "1.Ввести массив" << endl;

cout << "2.Вывести массив" << endl;

cout << "3.Метод линейного поиска" << endl;

cout << "4.Метод двоичного поиска" << endl;

cout << "5.Метод сортировки выбором" << endl;

cout << "6.Метод сортировки пузырьком" << endl;

cout << "7.Метод сортировки включением" << endl;

cout << "8.Метод быстрой сортировки" << endl;

cout << "9.Выход" << endl;

cin >> h;

switch (h)

{

case 1:

{

cout << "Введите количество элементов: ";

cin >> n;

while (n > 9999)

{

cout << "Введите элементы массива: ";

cin >> n;

}

mas = new int[n];

Input(n, mas);

break;

}

case 2:

{

Output(n, mas);

break;

}

case 3:

{

LineSearch(n, mas);

break;

}

case 4:

{

int value;

cout << "Искомое значение: " << endl;

cin >> value;

int pos = BinarSearch(value, n, mas);

if (value == mas[pos])

cout << "Позиция : " << pos + 1 << endl;

else

cout << "Не найдено." << endl;

break;

}

case 5:

{

Output(n, mas);

SelectSort(n, mas);

Output(n, mas);

break;

}

case 6:

{

Output(n, mas);

BubbleSort(n, mas);

Output(n, mas);

break;

}

case 7:

{

Output(n, mas);

IncludeSort(n, mas);

Output(n, mas);

break;

}

case 8:

{

Output(n, mas);

QuickSort(mas, 0, n - 1);

Output(n, mas);

break;

}

case 9:

{

break;

}

default:

{

cout << "Пункт меню выбран не правильно! Выберите подходящий пункт 1-9" << endl;

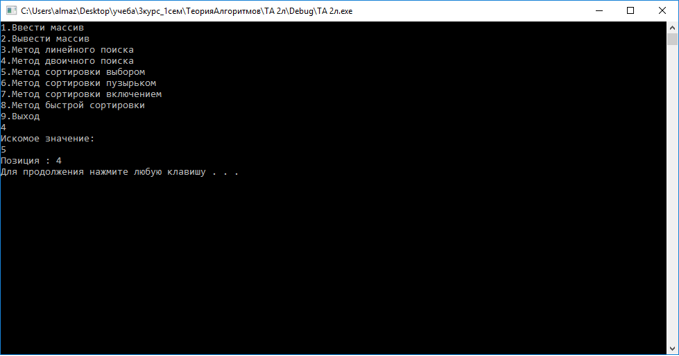
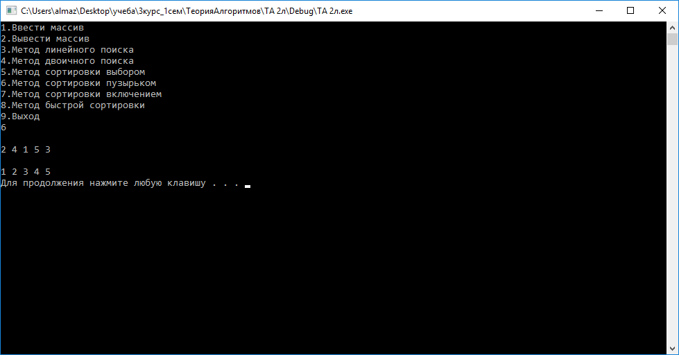
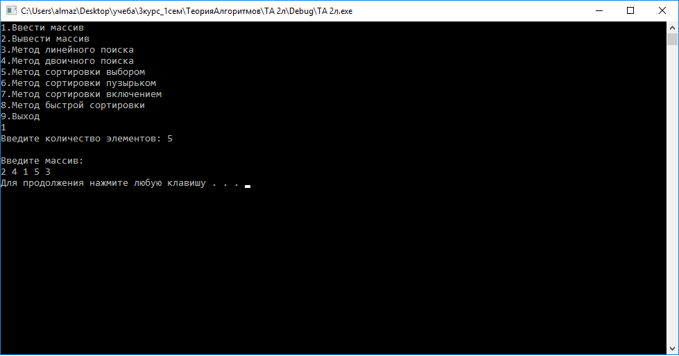
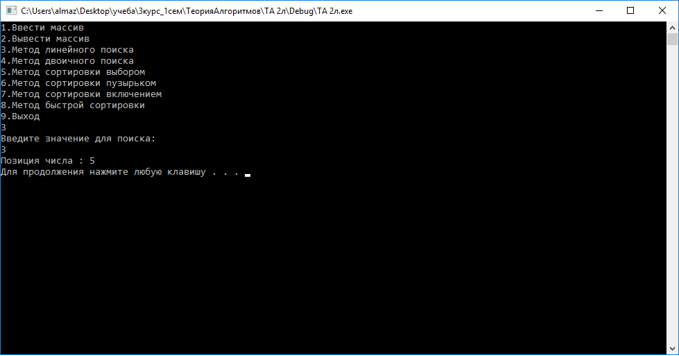
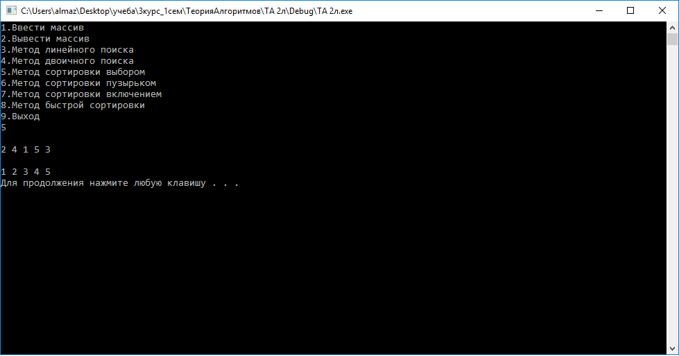
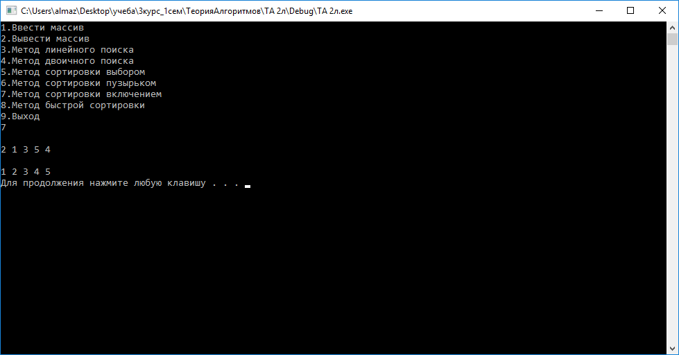
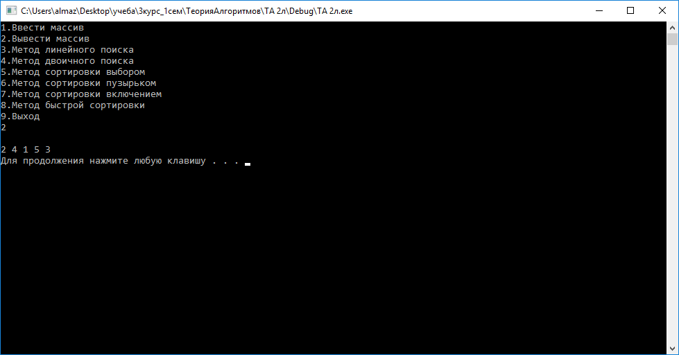
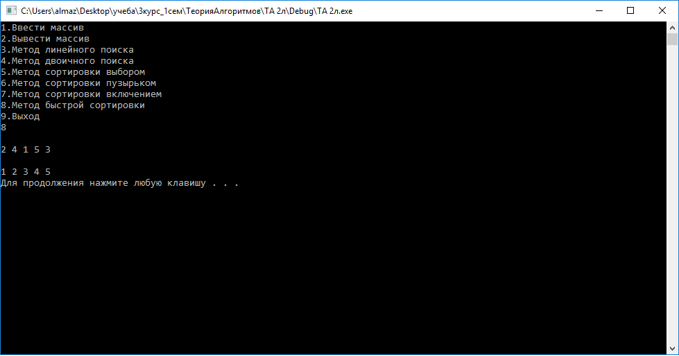
}

}

system("pause");

} while (h != 9);

}



Лабораторная работа 3.

#include <iostream>

#include <vector>

#include <list>

using namespace std;

//Хасанов Алмаз 4333

void Input(int n, int \*mas)

{

cout << endl << "Введите массив: " << endl;

for (int i = 0; i < n; i++)

cin >> mas[i];

}

void Output(int n, int \*mas)

{

cout << endl;

for (int i = 0; i < n; i++)

cout << mas[i] << " ";

cout << endl;

}

void ShellSort(int n, int \*mas)

{

int d = n / 2;

while (d != 0) {

for (int i = d; i < n; ++i) {

int tmp = mas[i];

int j;

for (j = i - d; j >= 0 && mas[j] > tmp; j -= d)

mas[j + d] = mas[j];

mas[j + d] = tmp;

}

d /= 2;

}

}

void iswap(int &n1, int &n2)

{

int temp = n1;

n1 = n2;

n2 = temp;

}

void HeapSort(int n, int \*mas)

{

int sh = 0;

bool b = false;

for (;;)

{

b = false;

for (int i = 0; i < n; i++)

{

if (i \* 2 + 2 + sh < n)

{

if ((mas[i + sh] > mas[i \* 2 + 1 + sh]) || (mas[i + sh] > mas[i \* 2 + 2 + sh]))

{

if (mas[i \* 2 + 1 + sh] < mas[i \* 2 + 2 + sh])

{

iswap(mas[i + sh], mas[i \* 2 + 1 + sh]);

b = true;

}

else if (mas[i \* 2 + 2 + sh] < mas[i \* 2 + 1 + sh])

{

iswap(mas[i + sh], mas[i \* 2 + 2 + sh]);

b = true;

}

}

if (mas[i \* 2 + 2 + sh] < mas[i \* 2 + 1 + sh])

{

iswap(mas[i \* 2 + 1 + sh], mas[i \* 2 + 2 + sh]);

b = true;

}

}

else if (i \* 2 + 1 + sh < n)

{

if (mas[i + sh] > mas[i \* 2 + 1 + sh])

{

iswap(mas[i + sh], mas[i \* 2 + 1 + sh]);

b = true;

}

}

}

if (!b) sh++;

if (sh + 2 == n) break;

}

}

int main()

{

int h, \*mas = new int[1], n;

setlocale(LC\_ALL, "Russian");

do

{

system("cls");

cout << "1.Ввести массив" << endl;

cout << "2.Вывести массив" << endl;

cout << "3.Метод сортировки Шела" << endl;

cout << "4.Пирамидальная сортировка" << endl;

cout << "5.Выход" << endl;

cin >> h;

switch (h)

{

case 1:

{

cout << "Введите количество элементов: ";

cin >> n;

while (n > 9999)

{

cout << "Введите элементы массива: ";

cin >> n;

}

mas = new int[n];

Input(n, mas);

break;

}

case 2:

{

Output(n, mas);

break;

}

case 3:

{

Output(n, mas);

ShellSort(n, mas);

Output(n, mas);

break;

}

case 4:

{

Output(n, mas);

HeapSort(n, mas);

Output(n, mas);

break;

}

case 5:

{

break;

}

default:

{

cout << "Пункт меню выбран не правильно! Выберите подходящий пункт 1-5" << endl;

}

}

system("pause");

} while (h != 5);

}

//////////////////////////////////////////////

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace TA3L

{

class Program

{

static void BucketSort(int[] mas)

{

List<int>[] lst = new List<int>[mas.Length];

for (int i = 0; i < lst.Length; ++i)

lst[i] = new List<int>();

int min = mas[0];

int max = mas[0];

for (int i = 1; i < mas.Length; ++i)

{

if (mas[i] < min)

min = mas[i];

else if (mas[i] > max)

max = mas[i];

}

double range = max - min;

for (int i = 0; i < mas.Length; ++i)

{

int bcktIdx = (int)Math.Floor((mas[i] - min) / range \* (lst.Length - 1));

lst[bcktIdx].Add(mas[i]);

}

for (int i = 0; i < lst.Length; ++i)

lst[i].Sort();

int id = 0;

for (int i = 0; i < lst.Length; ++i)

{

for (int j = 0; j < lst[i].Count; ++j)

mas[id++] = lst[i][j];

}

}

static void Main()

{

Console.WriteLine("Введите размер массива:");

int n = Convert.ToInt32(Console.ReadLine());

int[] mas = new int[n];

Console.WriteLine("Введите массив:");

for (int i = 0; i < mas.Length; ++i)

mas[i] = Convert.ToInt32(Console.ReadLine()); ;

Console.WriteLine(String.Join(" ", mas));

BucketSort(mas);

Console.WriteLine(String.Join(" ", mas));

Console.ReadKey();

}

}

}

//////////////////////////////////////////////

using System;

using System.Collections.Generic;

using System.Collections;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Porazryad

{

class Program

{

public static void RadixSorting(int[] mas, int range, int n)

{

ArrayList[] lists = new ArrayList[range];

for (int i = 0; i < range; ++i)

lists[i] = new ArrayList();

for (int step = 0; step < n; ++step)

{

for (int i = 0; i < mas.Length; ++i)

{

int a = (mas[i] % (int)Math.Pow(range, step + 1)) /

(int)Math.Pow(range, step);

lists[a].Add(mas[i]);

}

int k = 0;

for (int i = 0; i < range; ++i)

{

for (int j = 0; j < lists[i].Count; ++j)

{

mas[k++] = (int)lists[i][j];

}

}

for (int i = 0; i < range; ++i)

lists[i].Clear();

}

}

static void Main(string[] args)

{

Console.WriteLine("Введите длину массива:");

int n = Convert.ToInt32(Console.ReadLine());

int[] mas = new int[n];

Console.WriteLine("Введите массив:");

for (int i = 0; i < mas.Length; ++i) {

mas[i] = Convert.ToInt32(Console.ReadLine());

}

foreach(double x in mas) {

Console.Write(x + " ");

}

RadixSorting(mas, 10, 2);

Console.WriteLine(" ");

foreach (double x in mas) {

Console.Write(x + " ");

}

System.Console.ReadKey();

}

}

}

/////////////////////////////////////////////////////////////

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace TA3L

{

class Program

{

static void BucketSort(int[] mas)

{

List<int>[] lst = new List<int>[mas.Length];

for (int i = 0; i < lst.Length; ++i)

lst[i] = new List<int>();

int min = mas[0];

int max = mas[0];

for (int i = 1; i < mas.Length; ++i)

{

if (mas[i] < min)

min = mas[i];

else if (mas[i] > max)

max = mas[i];

}

double range = max - min;

for (int i = 0; i < mas.Length; ++i)

{

int bcktIdx = (int)Math.Floor((mas[i] - min) / range \* (lst.Length - 1));

lst[bcktIdx].Add(mas[i]);

}

for (int i = 0; i < lst.Length; ++i)

lst[i].Sort();

int id = 0;

for (int i = 0; i < lst.Length; ++i)

{

for (int j = 0; j < lst[i].Count; ++j)

mas[id++] = lst[i][j];

}

}

static void Main()

{

Console.WriteLine("Введите размер массива:");

int n = Convert.ToInt32(Console.ReadLine());

int[] mas = new int[n];

Console.WriteLine("Введите массив:");

for (int i = 0; i < mas.Length; ++i)

mas[i] = Convert.ToInt32(Console.ReadLine()); ;

Console.WriteLine(String.Join(" ", mas));

BucketSort(mas);

Console.WriteLine(String.Join(" ", mas));

Console.ReadKey();

}

}

}

/////////////////////////////////////////////////////////

