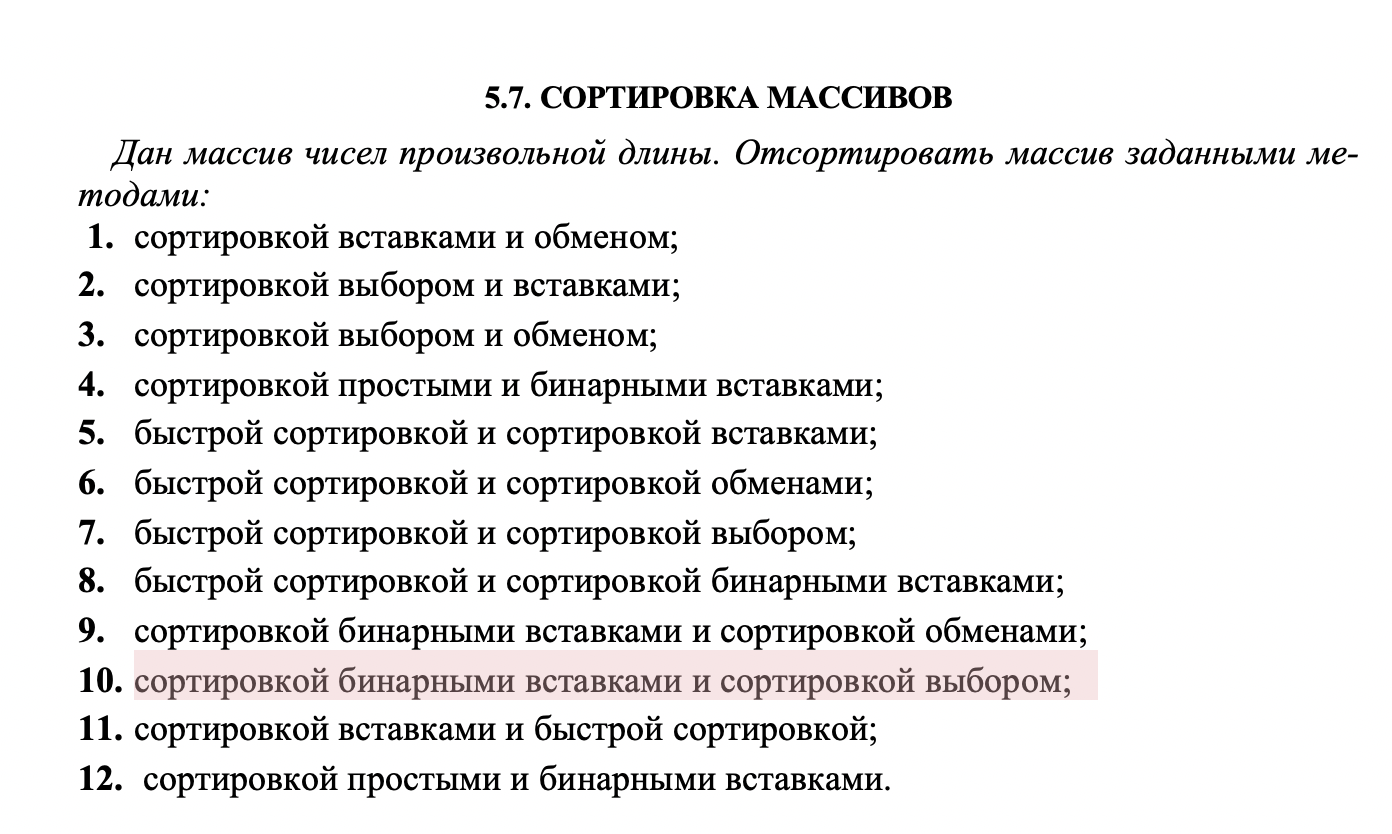
5.7



MAIN.CPP

#include <iostream>

#include <ctime>

#include "Source.cpp"

using namespace std;

int main()

{

srand(time(0));

const int size = 10;

int Iarray[size];

char Carray[size];

double Darray[size];

float Farray[size];

Enter\_random(Iarray, size);

cout << endl;

Enter\_random(Darray, size);

cout<<"\n --------------------------------------------------- \n" << endl;

BinaryInsertionSort(Iarray, size);

choicesSort(Iarray, size);

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

{

cout << setw(3) << Iarray[i];

}

cout<<"\n\n\n";

}

SOURCE.CPP

#include <iostream>

#include <time.h>

#include <iomanip>

using namespace std;

template < typename T >

void Enter\_random(T\* arr, int size)

{

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

{

int k = rand() % 20;

arr[i] = static\_cast<T>(k);

cout << setw(4) << arr[i] ;

}

}

template < typename T >

int binarySearch(T arr[], T item, T low, T high) {

if (high <= low)

return (item > arr[low])? (low + 1): low;

int mid = (low + high)/2;

if(item == arr[mid])

return mid+1;

if(item > arr[mid])

return binarySearch(arr, item, mid+1, high);

return binarySearch(arr, item, low, mid-1);

}

template < typename T >

void BinaryInsertionSort(T arr[], int n) {

int counter=0;

int i, loc, j, selected;

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

j = i - 1;

selected = arr[i];

loc = binarySearch(arr, selected, 0, j);

while (j >= loc) {

arr[j+1] = arr[j];

j--;

counter++;

}

arr[j+1] = selected;

}

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

{

cout <<" "<< arr[i];

}

cout <<"\n Counter "<< counter << endl;

}

template < typename T >

void choicesSort(T arrayPtr, int length\_array) // сортировка выбором

{

int counter=0;

for (int repeat\_counter = 0; repeat\_counter < length\_array; repeat\_counter++)

{

int temp = arrayPtr[0]; // временная переменная для хранения значения перестановки

for (int element\_counter = repeat\_counter + 1; element\_counter < length\_array; element\_counter++)

{

if (arrayPtr[repeat\_counter] > arrayPtr[element\_counter])

{

temp = arrayPtr[repeat\_counter];

arrayPtr[repeat\_counter] = arrayPtr[element\_counter];

arrayPtr[element\_counter] = temp;

counter++;

}

}

}cout <<"\n Counter "<< counter << endl;

}

Это я даже помню, посему вопросов нет

