**Московский государственный технический  
университет им. Н. Э. Баумана**

Факультет «Информатика и системы управления»  
Кафедра ИУ5 «Системы обработки информации и управление»

Курс «Основы программирования»

Отчет по лабораторной работе №5  
«**Сортировка одномерного числового массива**»

Выполнил: Проверил:

Студент группы ИУ5-11Б Преподаватель каф. ИУ5

Карпов Д.К. Аксенова М.В.  
Подпись и дата: Подпись и дата:

Москва, 2018 г.

**Задание**

Отсортировать числовой массив методом выбора максимального (минимального) элемента и методом пузырькового всплытия. По окончании сортировки вывести отсортированный массив и количество сделанных сравнений и перестановок элементов.

Сравнить быстродействие алгоритмов, которое определяется числом сравнений и перестановок, для исходного не отсортированного массива и для исходного массива, отсортированного в прямом и обратном порядке.

Исследовать зависимость быстродействия от размера массива. Возможность изменения длины массива реализуйте с помощью динамического массива, а для его инициализации используйте датчик случайных чисел (см. Приложение 1).

**Таблица переменных основной программы**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Назначение переменной** | **Идентификатор** | **Тип** | **Диапазон значений** | **Примечания** |
| Размерность входного массива | n | int | 5, 50, 500 | Входные данные |
| Рабочий массив | a | Int[n] | -99…99 | Входные и выходные данные |
| Рабочий массив | b | Int[n] | -99…99 | Входные и выходные данные |
| Счетчик цикла | i | int | 0…n-1 |  |
| Кол-во сравнений | nsrav | Int |  | Выходные данные |
| Кол-во перестановок | nper | int |  | Выходные данные |

**Переменные функций “Puzirek vozr” и “Puzirek ubyv”**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Назначение переменной** | **Идентификатор** | **Тип** | **Диапазон значений** | **Примечания** |
| Рабочий массив | a | Int[n] | -99…99 | Входные и выходные данные |
| Счетчик цикла | i | int | 0…n-1 | Внутренняя переменная |
| Счетчик цикла | j | int | 0…n-1 | Внутренняя переменная |
| Буферная переменная | tmp | int | -99…99 | Внутренняя переменная |
| Кол-во сравнений | nsrav | int |  | Выходные данные |
| Кол-во перестановок | nper | int |  | Выходные данные |
| Системное время начало работы программы | starttime | int |  | Внутренняя переменная |
| Системное время конца работы программы | endtime | int |  | Внутренняя переменная |

**Переменные функции “Sortir vozr”**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Назначение переменной** | **Идентификатор** | **Тип** | **Диапазон значений** | **Примечания** |
| Рабочий массив | a | Int[n] | -99…99 | Входные и выходные данные |
| Счетчик цикла | i | int | 0…n-1 | Внутренняя переменная |
| Счетчик цикла | j | int | 0…n-1 | Внутренняя переменная |
| Минимум | min | int | -99…99 | Внутренняя переменная |
| Буферная переменная | tmp | int | -99…99 | Внутренняя переменная |
| Кол-во сравнений | nsrav | int |  | Выходные данные |
| Кол-во перестановок | nper | int |  | Выходные данные |
| Системное время начало работы программы | starttime | int |  | Внутренняя переменная |
| Системное время конца работы программы | endtime | int |  | Внутренняя переменная |

**Переменные функции “Sortir ubyv”**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Назначение переменной** | **Идентификатор** | **Тип** | **Диапазон значений** | **Примечания** |
| Рабочий массив | a | Int[n] | -99…99 | Входные и выходные данные |
| Счетчик цикла | i | int | 0…n-1 | Внутренняя переменная |
| Счетчик цикла | j | int | 0…n-1 | Внутренняя переменная |
| Максимум | max | int | -99…99 | Внутренняя переменная |
| Буферная переменная | tmp | int | -99…99 | Внутренняя переменная |
| Кол-во сравнений | nsrav | int |  | Выходные данные |
| Кол-во перестановок | nper | int |  | Выходные данные |
| Системное время начало работы программы | starttime | int |  | Внутренняя переменная |
| Системное время конца работы программы | endtime | int |  | Внутренняя переменная |



**Рисунок 1. Блок-схема главной программы. Начало**.



**Рисунок 2. Блок-схема главной программы. Окончание.**



**Рисунок 3. Блок инициализации исходных массивов**

****

**Рисунок 4. Блок вывода исходного массива**



**Рисунок 5. Функция сортировки массива методом пузырька по возрастанию (puzirek\_vozr)**



**Рисунок 6. Функция сортировки массива методом пузырька по убыванию (puzirek\_ubyv)**



**Рисунок 7. Функция сортировки массива методом выбора по возрастанию (sortir\_vozr)**



**Рисунок 8. Функция сортировки массива методом выбора по убыванию (sortir\_ubyv)**

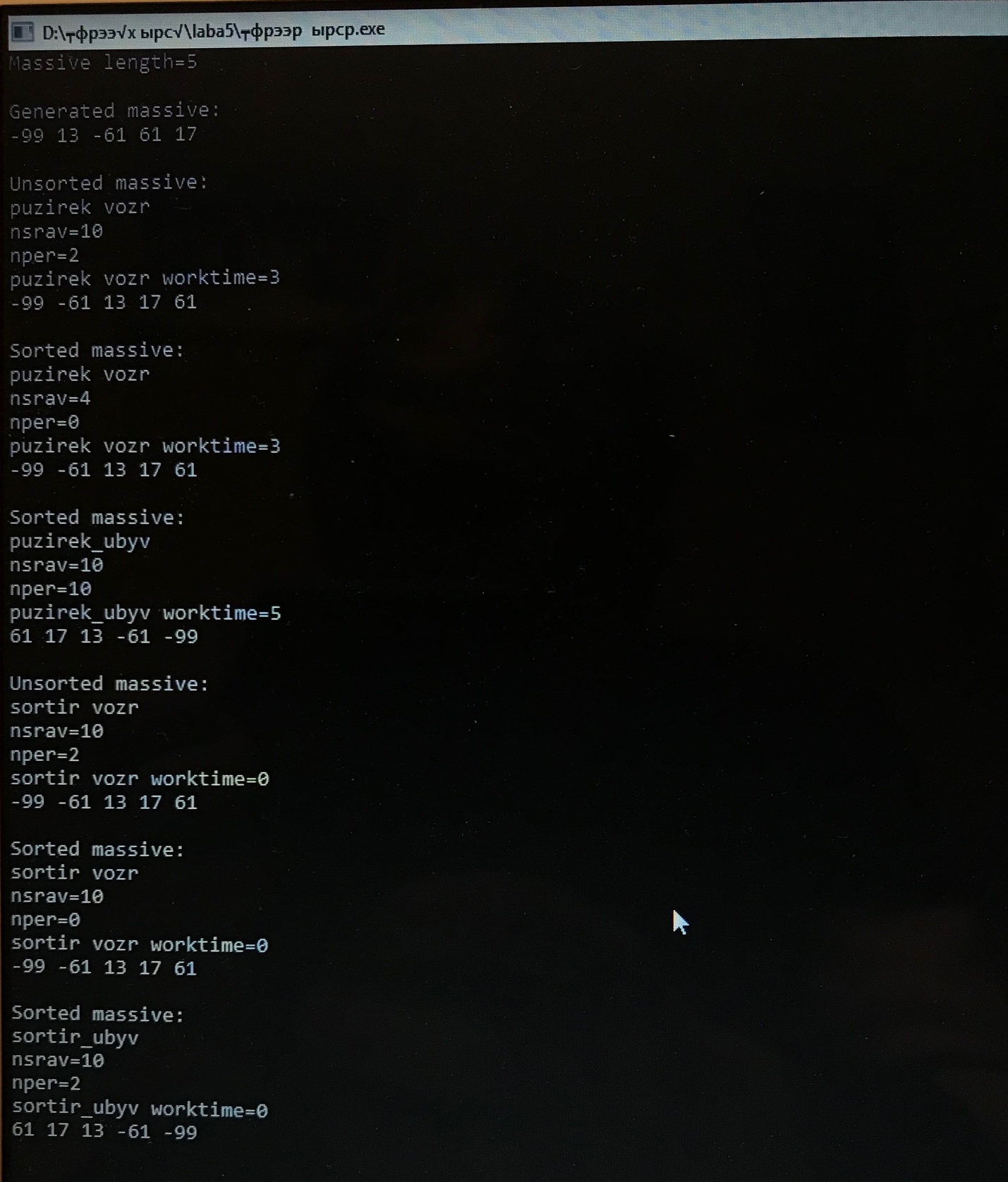
****

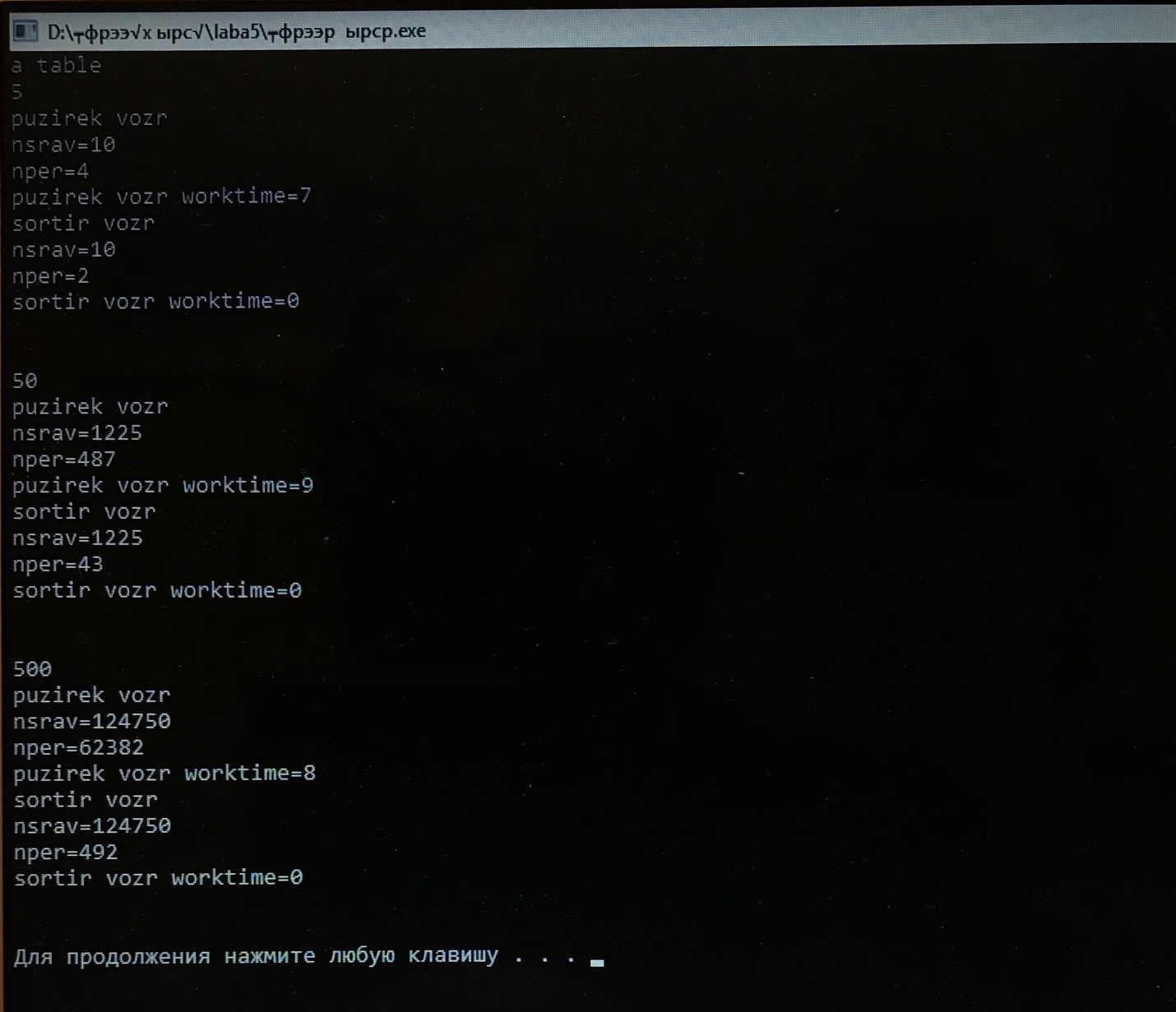
**Рисунок 9. Блок вывода результатов сортировки**

**Текст программы:**

|  |
| --- |
| #include <time.h>  #include <stdlib.h> // srand, rand  #include <iostream>  using namespace std;  int iRandom( int a, int b );  void puzirek\_vozr(int \*a,int n,int \*nsrav,int \*nper);  void puzirek\_ubyv(int \*a,int n,int \*nsrav,int \*nper);  void sortir\_vozr(int \*a,int n,int \*nsrav,int \*nper);  void sortir\_ubyv(int \*a,int n,int \*nsrav,int \*nper);  int main(int argc, char\*\* argv)  {  setlocale(LC\_ALL, "Russian");  int n = 5;  int i;  int nsrav, nper;  int \*a=new int[n];  int \*b=new int[n];  cout<<"Massive length="<<n<<endl;  cout<<endl;  cout<<"Generated massive:"<<endl;  for( i = 0; i < n; i++ )  { \*(a+i) =iRandom( -99, 99 );  \*(b+i)=\*(a+i);  cout<< \*(a+i)<<" ";  }  cout<<endl;  cout<<endl;  cout<<"Unsorted massive:"<<endl;  puzirek\_vozr(a,n,&nsrav,&nper);  cout<<"puzirek\_vozr"<<endl;  cout<<"nsrav="<<nsrav<<endl;  cout<<"nper="<<nper<<endl;  for( i = 0; i < n; i++ )  {  cout<< \*(a+i)<<" ";  }  cout<<endl;  cout<<endl;  cout<<"Sorted massive:"<<endl;  puzirek\_vozr(a,n,&nsrav,&nper);  cout<<"puzirek\_vozr"<<endl;  cout<<"nsrav="<<nsrav<<endl;  cout<<"nper="<<nper<<endl;  for( i = 0; i < n; i++ )  {  cout<< \*(a+i)<<" ";  }  cout<<endl;  cout<<endl;  cout<<"Sorted massive:"<<endl;  puzirek\_ubyv(a,n,&nsrav,&nper);  cout<<"puzirek\_ubyv"<<endl;  cout<<"nsrav="<<nsrav<<endl;  cout<<"nper="<<nper<<endl;  for( i = 0; i < n; i++ )  {  cout<< \*(a+i)<<" ";  }  cout<<endl;  cout<<endl;  cout<<"Unsorted massive:"<<endl;  sortir\_vozr(b,n,&nsrav,&nper);  cout<<"sortir\_vozr"<<endl;  cout<<"nsrav="<<nsrav<<endl;  cout<<"nper="<<nper<<endl;  for( i = 0; i < n; i++ )  {  cout<< \*(b+i)<<" ";  }  cout<<endl;  cout<<endl;  cout<<"Sorted massive:"<<endl;  sortir\_vozr(b,n,&nsrav,&nper);  cout<<"sortir\_vozr"<<endl;  cout<<"nsrav="<<nsrav<<endl;  cout<<"nper="<<nper<<endl;  for( i = 0; i < n; i++ )  {  cout<< \*(b+i)<<" ";  }  cout<<endl;  cout<<endl;  cout<<"Sorted massive:"<<endl;  sortir\_ubyv(b,n,&nsrav,&nper);  cout<<"sortir\_ubyv"<<endl;  cout<<"nsrav="<<nsrav<<endl;  cout<<"nper="<<nper<<endl;  for( i = 0; i < n; i++ )  {  cout<< \*(b+i)<<" ";  }  cout<<endl;  cout<<endl;  n=5;  for (int j=1; j<=3; j++)  { a=new int[n];  for( i = 0; i < n; i++ )  \*(a+i) =iRandom( -99, 99 );    cout<<"n="<<n<<endl;  puzirek\_vozr(a,n,&nsrav,&nper);  cout<<"puzirek nsrav="<<nsrav;  cout<<" nper="<<nper<<endl;    for( i = 0; i < n; i++ )  \*(a+i) =iRandom( -99, 99 );    sortir\_vozr(a,n,&nsrav,&nper);  cout<<"sortir nsrav="<<nsrav;  cout<<" nper="<<nper<<endl;  n=n\*10;  }  delete a; delete b;  system("pause");  return 0;  }  int iRandom( int a, int b )  {  return a + ( b - a +1) \* rand()/RAND\_MAX ;  }  void puzirek\_ubyv(int \*a,int n,int \*nsrav,int \*nper)  {  int i,j,tmp;  int starttime, endtime;  //int nsrav,nper;  \*nsrav=0;  \*nper=0;  starttime=clock();    for(i = 0 ; i < n - 1; i++)  {  for(j = 0 ; j < n - i - 1 ; j++)  {  \*nsrav=\*nsrav+1;  if( \*(a+j) < \*(a+j+1) )  {  tmp = \*(a+j);  \*(a+j) = \*(a+j+1) ;  \*(a+j+1)= tmp;  \*nper=\*nper+1;  }  }  }    endtime=clock();  //cout<<"puzirek\_ubyv worktime="<<endtime-starttime<<endl;  }  void sortir\_ubyv(int \*a,int n,int \*nsrav,int \*nper)  {  int i,j,max,tmp;  int starttime, endtime;  //int nsrav,nper;    \*nsrav=0;  \*nper=0;    starttime=clock();    for (int i = 0; i < n - 1; i++)  {  max = i;  for (int j = i + 1; j < n; j++)  {  \*nsrav=\*nsrav+1;  if (\*(a+j) > \*(a+max))  {  max= j;  }  }    if (max!= i) {  tmp = \*(a+i);  \*(a+i) = \*(a+max) ;  \*(a+max)= tmp;  \*nper=\*nper+1;    }  }  endtime=clock();  //cout<<"sortir\_ubyv worktime="<<(endtime-starttime)/CLK\_TCK<<endl;  }  void sortir\_vozr(int \*a,int n,int \*nsrav,int \*nper)  {  int i,j,min,tmp;  int starttime, endtime;  //int nsrav,nper;    \*nsrav=0;  \*nper=0;  starttime=clock();    for (int i = 0; i < n - 1; i++)  {  min = i;  for (int j = i + 1; j < n; j++)  {  \*nsrav=\*nsrav+1;  if (\*(a+j) < \*(a+min))  {  min= j;  }  }    if (min!= i) {  tmp = \*(a+i);  \*(a+i) = \*(a+min) ;  \*(a+min)= tmp;  \*nper=\*nper+1;    }  if (\*nper==0) break;  }  endtime=clock();  //cout<<"sortir vozr worktime="<<(endtime-starttime)/CLK\_TCK<<endl;  }  void puzirek\_vozr(int \*a,int n,int \*nsrav,int \*nper)  {  int i,j,tmp;  int starttime, endtime;  //int nsrav,nper;  \*nsrav=0;  \*nper=0;  starttime=clock();    for(i = 0 ; i < n - 1; i++)  {  for(j = 0 ; j < n - i - 1 ; j++)  {  \*nsrav=\*nsrav+1;  if( \*(a+j) > \*(a+j+1) )  {  tmp = \*(a+j);  \*(a+j) = \*(a+j+1) ;  \*(a+j+1)= tmp;  \*nper=\*nper+1;  }  }  if (\*nper==0) break;  }  endtime=clock();  //cout<<"puzirek vozr worktime="<<endtime-starttime<<endl;  } |

**Результат работы программы:**

****

****

**Выводы:**

Результаты тестирования прграммы:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Puzirek** | | **Sortir** | |
|  | **nsrav** | **nper** | **nsrav** | **nper** |
| N=5 | 10 | 4 | 10 | 2 |
| N=50 | 1225 | 478 | 1225 | 43 |
| N=500 | 124750 | 62382 | 124750 | 492 |

По результатам тестирования программы можно сделать вывод, что при увеличении размеров входного массива метод выбора является более эффективным, т.к. выполняет меньшее количество перестановок.