

Спецификация к Лабораторной работе №7

Упорядочение массива методами простого выбора и «пузырька».

(Вариант 22)

Абстракция A0

1. Постановка задачи.

Задание:

Написать программы упорядочения одномерного массива двумя методами:

- методом простого выбора и
- методом «пузырька».

Условие:

Заданный целочисленный массив $\{A_1, A_2, \dots, A_n\}$ отсортировать в порядке убывания (минимум - в конец).

Порядок элементов выводить на экран после каждого прохода по массиву.

2. Уточненная постановка задачи.

Дан одномерный целочисленный массив A из n элементов.

1) Выполнить сортировку методом простого выбора:

на каждом i-ом этапе сортировки поменять местами

- минимальный элемент массива $\{A_1, A_2, \dots, A_{n-i+1}\}$ с индексом k и
- элемент с индексом $(n - i + 1)$.

2) Выполнить сортировку методом "пузырька":

- на очередном этапе сравниваются все соседние элементы,
- в каждой паре элементы меняются местами, если первый элемент меньше второго,
- последний элемент полученного массива исключается из анализа на следующем этапе.

3. Пример с иллюстрацией.

```
In [5]: from IPython.display import Image
import numpy
import pylab
%matplotlib inline
pylab.style.use('seaborn-whitegrid')
```

```
In [2]: A1 = numpy.random.randint(-100, 100, size=20)
numpy.savetxt('A_Lab7_1.txt', A1, fmt='%5.0f')
As1 = numpy.sort(A1)[::-1]
str(list(A1)), str(list(As1))
```

```
Out[2]: (['-78, 26, -3, 32, -48, -97, -80, 49, 86, -32, 91, 58, 82, 19, 98, -65, -93, 40, 6, -47'],
['98, 91, 86, 82, 58, 49, 40, 32, 26, 19, 6, -3, -32, -47, -48, -65, -78, -80, -93, -97'])
```

```
In [21]: A2 = numpy.random.randint(0, 2, size=20)
numpy.savetxt('A_Lab7_2.txt', A2, fmt='%5.0f')
As2 = numpy.sort(A2)[::-1]
print('\n', A2, '\n', As2)
```

```
[1 1 0 0 0 0 0 1 1 1 1 1 0 0 1 0 0 1 0]
[1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0]
```

```
In [27]: A3 = numpy.random.randint(1, 10, size=5)
numpy.savetxt('A_Lab7_3.txt', A3, fmt='%5.0f')
As3 = numpy.sort(A3)[::-1]
print('\n', A3, '\n', As3)
```

```
[1 6 7 9 5]
[9 7 6 5 1]
```

```
In [30]: A4 = numpy.random.randint(-100, 0, size=10)
numpy.savetxt('A_Lab7_4.txt', A4, fmt='%5.0f')
As4 = numpy.sort(A4)[::-1]
print('\n', A4, '\n', As4)
```

```
[-75 -75 -11 -15 -7 -79 -86 -37 -65 -32]
[-7 -11 -15 -32 -37 -65 -75 -75 -79 -86]
```

```
In [32]: A5 = numpy.random.randint(0, 1000, size=10)
numpy.savetxt('A_Lab7_5.txt', A5, fmt='%5.0f')
As5 = numpy.sort(A5)[::-1]
print('\n', A5, '\n', As5)
```

```
[178 388 727 186 412 87 324 852 190 652]
[852 727 652 412 388 324 190 186 178 87]
```

```

In [22]: # A_Lab7_1.txt, A_Lab7_2.txt, A_Lab7_3.txt, A_Lab7_4.txt, A_Lab7_5.txt
# As_Lab7_1.txt, As_Lab7_2.txt, As_Lab7_3.txt, As_Lab7_4.txt, As_Lab7_5.txt

print(20*' ' + 'Лаб.7\n' + 90*'=' )

fin = str(input('Текстовый файл с элементами массива A: '))
fout = str(input('Текстовый файл с элементами массива A и элементами сортированного массива A: '))

with open(fin, 'r') as f:
    A = [int(x) for x in f.readlines()]

with open(fout, 'w') as f:
    f.write("{}\n".format(A))

method = str(input('Выберете метод сортировки массива A (simple или bubble): '))

print(90*'=' + '\n' + 'Сортировка массива A: \n')

def simple_sort(A):
    AA = []
    for j in range(len(A)-1):
        print(j)
        print(A)
        AA.extend(A)
        MIN, k = A[len(A)-j-1], -1
        for i in range(len(A)-j-2, -1, -1):
            if MIN > A[i]:
                MIN = A[i]
                k = i
        if k > -1:
            A[k] = A[len(A)-j-1]
            A[len(A)-j-1] = MIN
    return AA

def bubble_sort(A):
    AA = []
    for j in range(len(A)-1):
        print(A)
        AA.extend(A)
        c = 0
        for i in range(len(A)-j-1):
            if A[i] < A[i+1]:
                MIN = A[i]
                A[i] = A[i+1]
                A[i+1] = MIN
                c += 1
        if c < 1: break
    return AA

if method == 'simple':
    AA = simple_sort(A)
elif method == 'bubble':
    AA = bubble_sort(A)
else:
    print('Неправильный ввод названия метода сортировки')

with open(fout, 'w') as f:
    f.write("{}\n".format(A))

```

Лаб.7

```

=====
Текстовый файл с элементами массива A: A_Lab7_1.txt
Текстовый файл с элементами массива A и элементами сортированного массива A: As_Lab7_1.txt
Выберете метод сортировки массива A (simple или bubble): bubble
=====

```

Сортировка массива A:

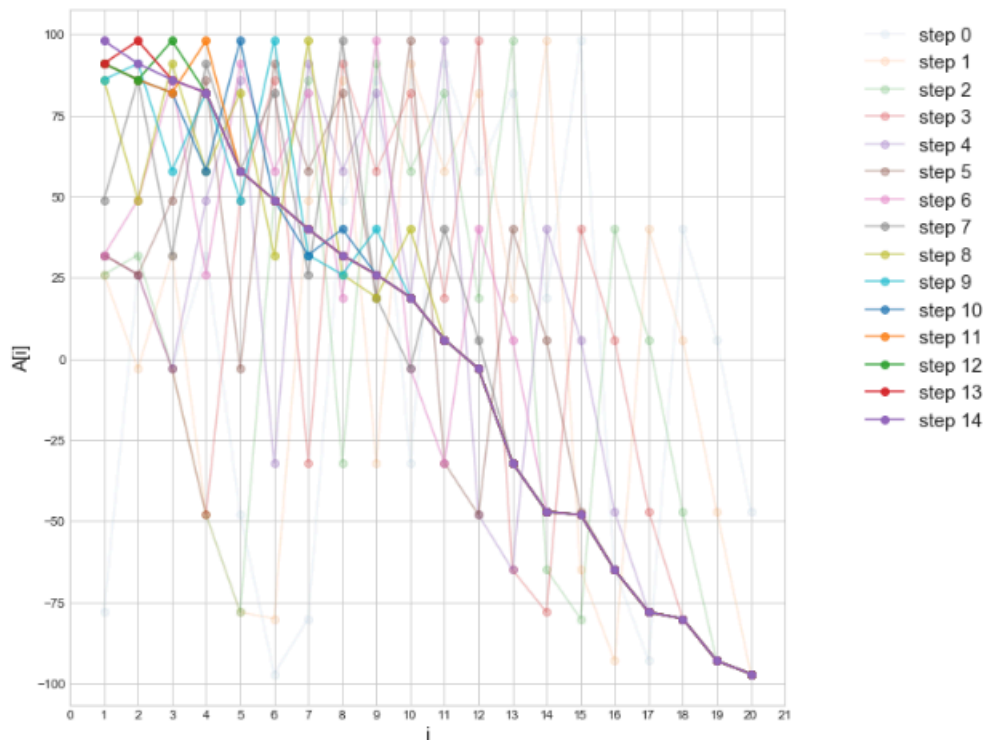
```

[-78, 26, -3, 32, -48, -97, -80, 49, 86, -32, 91, 58, 82, 19, 98, -65, -93, 40, 6, -47]
[26, -3, 32, -48, -78, -80, 49, 86, -32, 91, 58, 82, 19, 98, -65, -93, 40, 6, -47, -97]
[26, 32, -3, -48, -78, 49, 86, -32, 91, 58, 82, 19, 98, -65, -80, 40, 6, -47, -93, -97]
[32, 26, -3, -48, 49, 86, -32, 91, 58, 82, 19, 98, -65, -78, 40, 6, -47, -80, -93, -97]
[32, 26, -3, 49, 86, -32, 91, 58, 82, 19, 98, -48, -65, 40, 6, -47, -78, -80, -93, -97]
[32, 26, 49, 86, -3, 91, 58, 82, 19, 98, -32, -48, 40, 6, -47, -65, -78, -80, -93, -97]
[32, 49, 86, 26, 91, 58, 82, 19, 98, -3, -32, 40, 6, -47, -48, -65, -78, -80, -93, -97]
[49, 86, 32, 91, 58, 82, 26, 98, 19, -3, 40, 6, -32, -47, -48, -65, -78, -80, -93, -97]
[86, 49, 91, 58, 82, 32, 98, 26, 19, 40, 6, -3, -32, -47, -48, -65, -78, -80, -93, -97]
[86, 91, 58, 82, 49, 98, 32, 26, 40, 19, 6, -3, -32, -47, -48, -65, -78, -80, -93, -97]
[91, 86, 82, 58, 98, 49, 32, 40, 26, 19, 6, -3, -32, -47, -48, -65, -78, -80, -93, -97]
[91, 86, 82, 98, 58, 49, 40, 32, 26, 19, 6, -3, -32, -47, -48, -65, -78, -80, -93, -97]
[91, 86, 98, 82, 58, 49, 40, 32, 26, 19, 6, -3, -32, -47, -48, -65, -78, -80, -93, -97]
[91, 98, 86, 82, 58, 49, 40, 32, 26, 19, 6, -3, -32, -47, -48, -65, -78, -80, -93, -97]
[98, 91, 86, 82, 58, 49, 40, 32, 26, 19, 6, -3, -32, -47, -48, -65, -78, -80, -93, -97]

```

```
In [40]: pylab.figure(figsize=(10,10))
for i in range(int(len(AA)/20)):
    pylab.plot(range(20), AA[20*i:20*(i+1)], '-o',
               alpha=20*(i+0.8)/len(AA), label='step {}'.format(i))

pylab.xlabel('i', fontsize=15)
pylab.ylabel('A[i]', fontsize=15)
pylab.xticks(range(-1, len(A)+1), range(len(A)+2));
pylab.legend(fontsize=15, shadow=True, bbox_to_anchor=(1.3,1.0));
```



4. Таблица данных

Класс	Имя	Описание (смысл, диапазон, точность)	Тип	Структура	Формат в/в
входные данные	fin	название текстового файла с элементами массива A	символ	строка	"+XX.txt"
входные данные	fout	название текстового файла с элементами массива A	символ	строка	"+XX.txt"
входные данные	method	название метода сортировки массива A	символ	строка	"+XX"
входные данные	A	вводимые из текстового файла числа исходного массива	цел	одномерный массив	+XX (:4)
выходные данные	A	выводимые в текстовый файл числа сортированного массива	цел	одномерный массив	+XX (:4)
промежуточные данные	n	количество элементов массива	цел	простая переменная	---
промежуточные данные	MIN	минимальный элемент при сортировке	цел	простая переменная	---
промежуточные данные	m	индекс минимального элемент при сортировке	цел	простая переменная	---
промежуточные данные	i	индекс текущего элемента, $1 \leq i \leq 50$	цел	простая переменная	---
промежуточные данные	j	индекс текущего элемента, $1 \leq j \leq 50$	цел	простая переменная	---
промежуточные данные	c	подсчет количества перестановок элементов, $0 \leq c \leq 49$	цел	простая переменная	---

! в Delphi выход из цикла с 1 до 20 с шагом +1 происходит при значении 21, из цикла с 20 до 1 – при 0

5. Входная форма

- обр 1.1 Текстовый файл с элементами массива A: < fin >
- обр 1.2 Текстовый файл с элементами массива A и элементами сортированного массива A: < fout >
- обр 2 Выберите метод сортировки массива A (simple или bubble): < method >
- обр 3.1 Сортировка массива A:
- обр 3.2 < A[1] > < A[2] > ... < A[n] >

6. Выходная форма

- обр 4 Лаб. 7
- обр 5.1 Текстовый файл с элементами массива A: < fin >
- обр 5.2 Текстовый файл с элементами массива A и элементами сортированного массива A: < fout >
- обр 6 Выберите метод сортировки массива A (simple или bubble): < method >
- обр 7.1 Сортировка массива A:
- обр 7.2 < A[1] > < A[2] > ... < A[n] >
- ...
- < A[1] > < A[2] > ... < A[n] >

7. Аномалии

8. Функциональные тесты

Исходные данные	-	-	-	Результаты	Тест
fin	fout	method	A	A	№
A_Lab7_1.txt	As_Lab7_1.txt	bubble	[-78 26 -3 32 -48 -97 -80 49 86 -32 91 58 82 19 98 -65 -93 40 6 -47]	[98 91 86 82 58 49 40 32 26 19 6 -3 -32 -47 -48 -65 -78 -80 -93 -97]	1
-	-	-	-	-	-
A_Lab7_2.txt	As_Lab7_2.txt	simple	[1 1 0 0 0 0 0 1 1 1 1 1 0 0 1 0 0 1 0]	[1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0]	2
-	-	-	-	-	-
A_Lab7_3.txt	As_Lab7_3.txt	bubble	[1 6 7 9 5]	[9 7 6 5 1]	3
-	-	-	-	-	-
A_Lab7_4.txt	As_Lab7_4.txt	simple	[-75 -75 -11 -15 -7 -79 -86 -37 -65 -32]	[-7 -11 -15 -32 -37 -65 -75 -75 -79 -86]	4
-	-	-	-	-	-
A_Lab7_5.txt	As_Lab7_5.txt	bubble	[178 388 727 186 412 87 324 852 190 652]	[852 727 652 412 388 324 190 186 178 87]	5

№ теста	Входные данные	-	-	-	Ожидаемый результат	Смысл теста
-	-	-	-	-	-	-
1	fin = 'A_Lab7_1.txt'	fout = 'As_Lab7_1.txt'	method = 'bubble'	A = [-78 26 -3 32 -48 -97 -80 49 86 -32 91 58 82 19 98 -65 -93 40 6 -47]	A = [98 91 86 82 58 49 40 32 26 19 6 -3 -32 -47 -48 -65 -78 -80 -93 -97]	Подтвердить правильность расчетов для широкого спектра целочисленных значений
-	-	-	-	-	-	-
2	fin = 'A_Lab7_2.txt'	fout = 'As_Lab7_2.txt'	method = 'simple'	A = [1 1 0 0 0 0 0 1 1 1 1 1 1 0 0 1 0 0 1 0]	A = [1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0]	Протестировать простейший случай с бинарными значениями
-	-	-	-	-	-	-
3	fin = 'A_Lab7_3.txt'	fout = 'As_Lab7_3.txt'	method = 'bubble'	A = [1 6 7 9 5]	A = [9 7 6 5 1]	Протестировать случай сортировки натуральных чисел
-	-	-	-	-	-	-
4	fin = 'A_Lab7_4.txt'	fout = 'As_Lab7_4.txt'	method = 'simple'	A = [-75 -75 -11 -15 -7 -79 -86 -37 -65 -32]	A = [-7 -11 -15 -32 -37 -65 -75 -75 -79 -86]	Подтвердить правильность расчетов в случае отрицательных чисел
-	-	-	-	-	-	-
5	fin = 'A_Lab7_5.txt'	fout = 'As_Lab7_5.txt'	method = 'bubble'	A = [178 388 727 186 412 87 324 852 190 652]	A = [852 727 652 412 388 324 190 186 178 87]	Протестировать массив, состоящий из неотрицательных чисел

Метод	-	Результаты	№ теста
-	-	-	-
bubble	A	14 этапов сортировки, корректный результат	1
-	-	-	-
simple	A	19 этапов сортировки, корректный результат	2
-	-	-	-
bubble	A	3 этапа сортировки, корректный результат	3
-	-	-	-
simple	A	9 этапов сортировки, корректный результат	4
-	-	-	-
bubble	A	7 этапов сортировки, корректный результат	5

9. Метод

Отделим ввод-вывод от обработки данных и разделим задачу на три подзадачи:

1. Подзадача A 0.1. Запросить названия текстовых файлов ввода-вывода, считать из текстового файла ввода исходные данные (обр.1-3), затем вывести их на экран (обр.4-7.1) для визуального подтверждения.
2. Подзадача A 0.2. Решение поставленной задачи: на основе введенных исходных данных (A) сформировать сортированный одним из заданных методов (simple, bubble) массив (A)
3. Подзадача A 0.3. Ввести полученные результаты (A) на экран (обр.7.2) и записать в файл вывода исходный массив и полученный результат.

Ввод-вывод тривиален и представляет собой чисто техническую задачу.

Для осуществления процесса ввода-вывода и обработки массива потребуются промежуточные переменные:

Для осуществления процесса ввода-вывода и обработки массива потребуются промежуточные переменные:

- i – индекс текущего элемента массива,
- j – индекс текущего элемента массива,
- n – количество элементов в массиве,
- m – индекс минимального элемент при сортировке,
- MIN – минимальный элемент при сортировке,
- c – подсчет количества перестановок элементов.

Их следует добавить в таблицу данных.

Алгоритм необходим только для обработки входных данных и расчета результата.

10. Алгоритм

[Sample diagrams](#)

```
In [55]: %%file block_diagram

blockdiag {
    orientation = portrait;
    class start_end [shape = ellipse, fontsize = 15, color = lightskyblue];
    class input_out [shape = flowchart.input, fontsize = 15, width = 600, color = lightcyan];
    class main_box [shape = box, width = 320, height = 150, fontsize = 18];

    Начало ->
    "Вывод заголовка по обр.4 на экран" ->
    "Ввод имени файла по обр.1.1" -> "Ввод имени файла по обр.1.2" ->
    "Ввод названия метода по обр.2" ->
    "Вывод пояснения по обр.3.1" -> "Ввод А из файла по обр.3.2" ->
    "Вывод имени файла по обр.5.1 на экран" -> "Вывод имени файла по обр.5.2 на экран" ->
    "Вывод названия метода по обр.6" ->
    "Вывод пояснения по обр.7.1 на экран" -> "Сортировка массива выбранным методом" ->
    "Вывод А на экран и в файл по обр.7.2" ->
    Конеч;

    Начало, Конеч [class = "start_end"];
    "Ввод имени файла по обр.1.1", "Ввод имени файла по обр.1.2" [class = "input_out"];
    "Ввод названия метода по обр.2", "Вывод названия метода по обр.6" [class = "input_out"];
    "Вывод пояснения по обр.3.1", "Ввод А из файла по обр.3.2" [class = "input_out"];
    "Вывод имени файла по обр.5.1 на экран", "Вывод имени файла по обр.5.2 на экран" [class = "input_out"];
    "Вывод пояснения по обр.7.1 на экран", "Вывод А на экран и в файл по обр.7.2" [class = "input_out"];
    "Вывод заголовка по обр.4 на экран" [class = "input_out"];
    "Сортировка массива выбранным методом" [class = "main_box"];

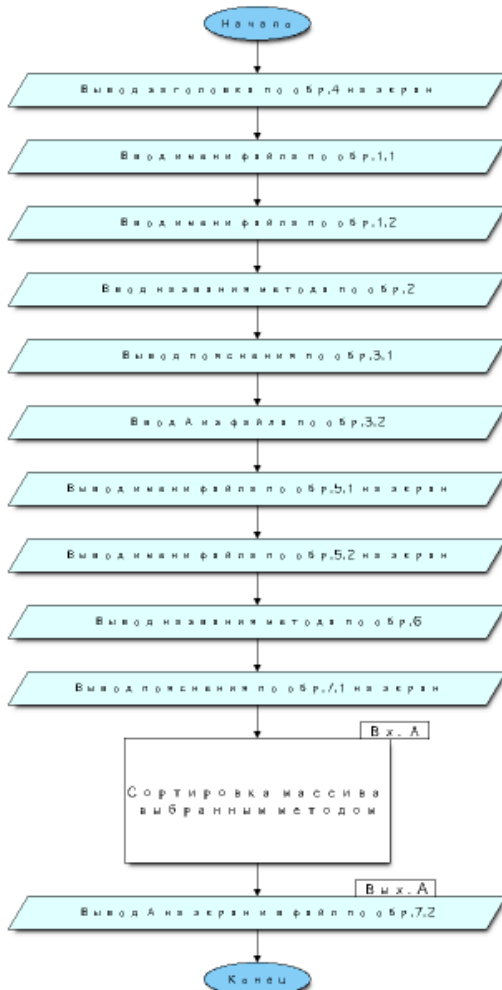
    "Вывод пояснения по обр.7.1 на экран" ->
    "Сортировка массива выбранным методом" [label = 'Вх. А', fontsize = 18];
    "Сортировка массива выбранным методом" ->
    "Вывод А на экран и в файл по обр.7.2" [label = 'Вых. А', fontsize = 18];
}
```

Overwriting block_diagram

```
In [56]: !blockdiag block_diagram
```

```
In [57]: Image("block_diagram.png")
```

Out[57]:



11. Программа на *Delphi*.

Диалоговый вариант (ввод названий файлов и данных из текстового файла, вывод на экран и в текстовый файл)

Синтаксис:

[Pascal Tutorial](#)

```
In [ ]: program Lab7; //сохранить как lab7.dpr (DPR  Delphi Project)
{$APPTYPE CONSOLE}
Uses Windows; // для русификации (сделать комментарием в онлайн-компиляторе!)
const
  {заглушка для теста 1}
  TA1: array[1..20] of integer = (98,91,86,82,58,49,40,32,26,19,6,-3,-32,-47,-48,-65,-78,-80,-93,-97);
  {заглушка для теста 2}
  TA2: array[1..20] of integer = (1,1,1,1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0);
  {заглушка для теста 3}
  TA3: array[1..5] of integer = (9,7,6,5,1);
  {заглушка для теста 4}
  TA4: array[1..10] of integer = (-7,-11,-15,-32,-37,-65,-75,-75,-79,-86);
  {заглушка для теста 5}
  TA5: array[1..10] of integer = (852,727,652,412,388,324,190,186,178,87);

var
  i: integer;
  fin, fout, method: string;
  tfin, tfout: TextFile;
  A: array of integer;

begin
  setConsoleCP(1251); // для ввода русификация и (сделать строку комментарием в онлайн-компиляторе!)
  setConsoleOutputCP(1251); // для вывода (сделать строку комментарием в онлайн-компиляторе!)

  {A0.1, A0.3 - ввод-вывод входных данных}
  writeln('Лаб.7' :40); {вывод заголовка с переходом на следующую строку по обр.4}

  {ввод-вывод строковой переменной - имени файла с входными данными по обр.1.1, 5.1}
  writeln('Текстовый файл с элементами массива A: ');
  readln(fin);
  writeln(fin);

  {ввод-вывод строковой переменной - имени файла с выходными данными по обр.1.2, 5.2}
  writeln('Текстовый файл с элементами массива A и элементами отсортированного массива A: ');
  readln(fout);
  writeln(fout);

  {ввод-вывод строковой переменной - названия метода сортировки по обр.2, 6}
  writeln('Выберете метод сортировки массива A (simple или bubble): ');
  readln(method);
  writeln(method);

  {ввод-вывод исходного массива A по обр.3.1, 3.2, 7.1, 7.2}
  writeln('Сортировка массива A: '); {вывод пояснения с переходом на следующую строку}

  AssignFile(tfin, fin); reset(tfin); {открыть файл для чтения}
  AssignFile(tfout, fout); rewrite(tfout); {открыть файл для записи}

  i := 1;

  while not eof(tfin) do
  begin
    SetLength(A,i);
    readln(tfin, A[i]); {ввод элемента и переход на следующую строку}
    writeln(A[i]:4); {вывод элемента и переход на следующую строку на экране}
    writeln(tfout, A[i]:4); {вывод элемента и переход на следующую строку в файле}
    i := i + 1;
  end;

  CloseFile(tfin); {закрыть файл для чтения}

  for i:=1 to 80 do
    write('='); {отделим визуально чертой и строкой введенные и выводимые значения}
  writeln;

  for i:=1 to Length(TA1) do
  begin
    writeln(TA1[i]:4); {вывод элемента и переход на следующую строку на экране}
    writeln(tfout, TA1[i]:4); {вывод элемента и переход на следующую строку в файле}
  end;

  CloseFile(tfout); {закрыть файл для записи}

end.
```

Вариант программы для онлайн-компилятора.

[Compile and Execute Pascal Online](#)

```
In [ ]: // main.pas
program Lab7; // сохранить как lab7.dpr (DPR  Delphi Project)
const
  {заглушка для теста 1}
  TA1: array[1..20] of integer = (98,91,86,82,58,49,40,32,26,19,6,-3,-32,-47,-48,-65,-78,-80,-93,-97);

var
  i: integer;
  A: array of integer;
  method: string;

begin
```

```

begin

    {A0.1, A0.3 – ввод-вывод входных данных}
    writeln('Лаб.7' :40); {вывод заголовка с переходом на следующую строку по обр.4}

    {ввод-вывод строковой переменной – названия метода сортировки по обр.2, 6}
    writeln('Выберете метод сортировки массива A (simple или bubble): ');
    readln(method);
    writeln(method);

    {ввод-вывод исходного массива A по обр.3.1, 3.2, 7.1, 7.2}
    writeln('Сортировка массива A: '); {вывод пояснения с переходом на следующую строку}

    i := 2;

    while not eoln do
    begin
        SetLength(A,i);
        readln(A[i]); {ввод элемента и переход на следующую строку}
        writeln(A[i]:4); {вывод элемента и переход на следующую строку на экране}
        i := i + 1;
    end;

    for i:=1 to 80 do
        write(' '); {отделим визуально чертой и строкой введенные и выводимые значения}
    writeln;

    for i:=1 to Length(TA1) do
    begin
        writeln(TA1[i]:4); {вывод элемента и переход на следующую строку на экране}
    end;
end.

```

```

]: // STDIN
bubble
-78
26
-3
32
-48
-97
-80
49
86
-32
91
58
82
19
98
-65
-93
40
6
-47

```

Раскрытие абстракции A0.2

Поскольку условие всей задачи совпадает (за исключением необходимости ввода-вывода) с условием выделенной подзадачи, вместо полной спецификации с пунктами 1-11 выполнено просто дополнение к пунктам 9-11.

9. Метод

Пусть

- i – номер текущий точки,
- j – номер текущий точки,
- n – количество элементов в массиве,
- m – индекс минимального элемент при сортировке,
- MIN – минимальный элемент при сортировке,
- s – переменная для подсчета количества перестановок элементов,
- $A[i] \vee A[j]$ – соответствующий элемент исходного массива A ,
- $method$ – название метода сортировки массива A .

При выполнении программы выбирается метод сортировки массива.

Элементы сортируются в цикле, результат каждого этапа выводится на экран.

При сортировке методом *простого выбора*:

на каждом i -ом этапе сортировки меняются местами

- минимальный элемент массива $\{A_1, A_2, \dots, A_{n-i+1}\}$ с индексом m и
- элемент с индексом $(n - i + 1)$.

При сортировке методом "пузырька":

- на очередном этапе сравниваются все пары соседних элементов,
- в каждой паре элементы меняются местами, если первый элемент меньше второго,
- последний элемент полученного массива исключается из анализа на следующем этапе.

10. Алгоритм

Метод простого выбора

```
In [10]: %%file block_diagram2

blockdiag {
    orientation = portrait;

    class yes [thick, label = "Да", color = blue, textcolor = blue];
    class no [thick, label = "Нет", color = red, textcolor = red];
    class start_end [shape = ellipse, fontsize = 18, color = lightskyblue];
    class input_out [shape = flowchart.input, fontsize = 18,
        width = 400, color = lightcyan];
    class loop_in [shape = flowchart.loopin, fontsize = 18,
        width = 200, color = lightsteelblue];
    class loop_out [shape = flowchart.loopout, fontsize = 18,
        width = 200, color = lightsteelblue];
    class condition [shape = flowchart.condition, fontsize = 18,
        width = 250, height = 100, color = lightsteelblue];
    class command_box [shape = roundedbox, fontsize = 18, width = 300];

    Начало -> "j := 0; +1; n-1" -> "MIN := 10000" ->
    "i := 1; +1; n-j" -> "A[i] < MIN" -> "MIN := A[i]; m := i" ->
    "i = n-j" -> "A[m] := A[n-j]; A[n-j] := MIN" ->
    "j = n-1" -> Конц;
    "A[i] < MIN" -> "i = n-j";

    Начало, Конц [class = "start_end"];
    "j := 0; +1; n-1", "i := 1; +1; n-j" [class = "loop_in"];
    "i = n-j", "j = n-1" [class = "loop_out"];
    "A[i] < MIN" [class = "condition"];
    "MIN := 10000", "MIN := A[i]; m := i" [class = "command_box"];
    "A[m] := A[n-j]; A[n-j] := MIN" [class = "command_box"];

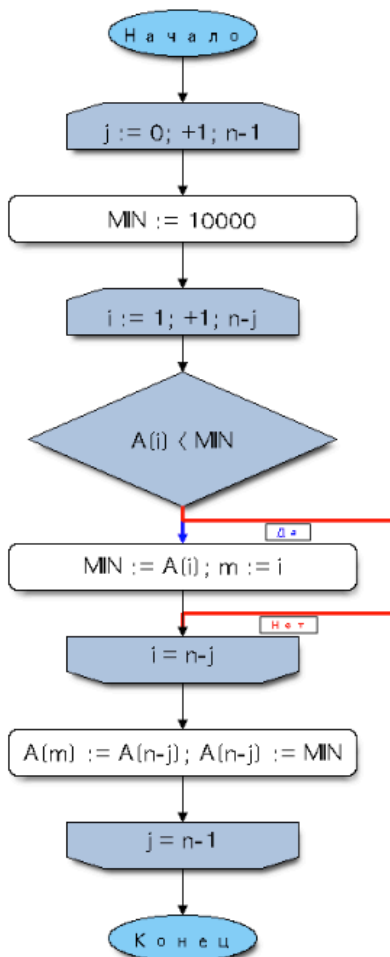
    "A[i] < MIN" -> "MIN := A[i]; m := i" [class = 'yes'];
    "A[i] < MIN" -> "i = n-j" [class = 'no'];
}
```

Overwriting block_diagram2

```
In [11]: lblockdiag block_diagram2
```

```
In [12]: Image("block_diagram2.png")
```

Out[12]:



Метод "пузырька"

```
In [16]: %%file block_diagram2

blockdiag {
    orientation = portrait;

    class yes {thick, label = "Да", color = blue, textcolor = blue};
    class no {thick, label = "Нет", color = red, textcolor = red};
    class start_end [shape = ellipse, fontsize = 18, color = lightsteelblue];
    class input_out [shape = flowchart.input, fontsize = 18,
        width = 400, color = lightcyan];
    class loop_in [shape = flowchart.loopin, fontsize = 18,
        width = 200, color = lightsteelblue];
    class loop_out [shape = flowchart.loopout, fontsize = 18,
        width = 200, color = lightsteelblue];
    class condition [shape = flowchart.condition, fontsize = 18,
        width = 250, height = 100, color = lightsteelblue];
    class command_box [shape = roundedbox, fontsize = 18, width = 300];

    Начало -> "j := 0; +1; n-1" -> "c := 0" ->
    "i := 2; +1; n-j" -> "A[i-1] < A[i]" ->
    "MIN := A[i-1]; A[i-1] := A[i]" -> "A[i] := MIN; c := c + 1" ->
    "i = n-j" -> "c = 0" ->
    "j = n-1" -> Конеч;
    "A[i-1] < A[i]" -> "i = n-j";
    "c = 0" -> Конеч;

    Начало, Конеч [class = "start_end"];
    "j := 0; +1; n-1", "i := 2; +1; n-j" [class = "loop_in"];
    "i = n-j", "j = n-1" [class = "loop_out"];
    "A[i-1] < A[i]", "c = 0" [class = "condition"];
    "c := 0", "MIN := A[i-1]; A[i-1] := A[i]" [class = "command_box"];
    "A[i] := MIN; c := c + 1" [class = "command_box"];

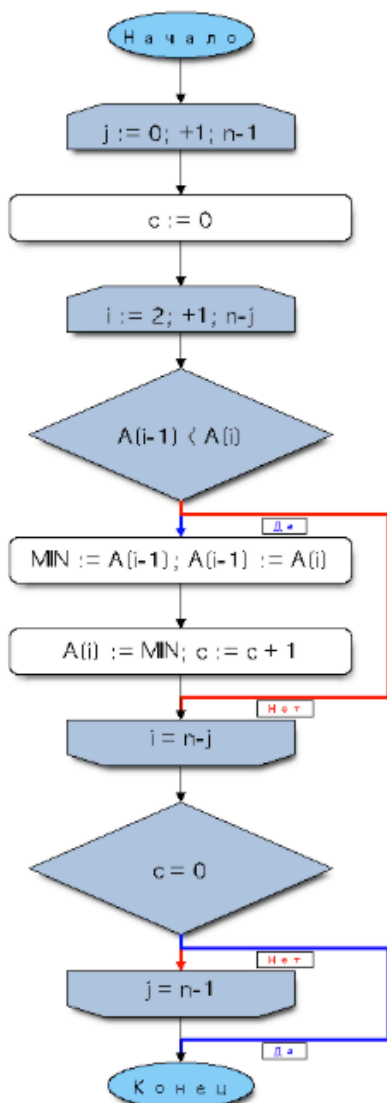
    "A[i-1] < A[i]" -> "MIN := A[i-1]; A[i-1] := A[i]" [class = 'yes'];
    "A[i-1] < A[i]" -> "i = n-j" [class = 'no'];
    "c = 0" -> Конеч [class = 'yes'];
    "c = 0" -> "j = n-1" [class = 'no'];
}
```

Overwriting block_diagram2

```
In [17]: !blockdiag block_diagram2
```

```
In [18]: Image("block_diagram2.png")
```

Out[18]:



11. Программа на *Delphi*.

Программный код раскрытия абстракции.


```
In [ ]: if method = 'simple' then
begin
  for j := 0 to n-1 do
  begin
    MIN := 10000;
    for i := 1 to n-j do
    begin
      write(A[i]:4); {вывод элемента на экран}
      if A[i] < MIN then
      begin
        MIN := A[i];
        m := i;
      end;
    end;
    A[m] := A[n-j]; A[n-j] := MIN;
    writeln;
  end;
end;

if method = 'bubble' then
begin
  for j := 0 to n-1 do
  begin
    c := 0;
    write(A[1]:4);
    for i := 2 to n-j do
    begin
      write(A[i]:4); {вывод элемента на экран}
      if A[i-1] < A[i] then
      begin
        MIN := A[i-1];
        A[i-1] := A[i];
        A[i] := MIN;
        c := c + 1;
      end;
    end;
    writeln;
    if c = 0 then break;
  end;
end;
end;
```

Диалоговый вариант (ввод названий файлов и данных из текстового файла, вывод на экран и в текстовый файл)

Синтаксис:

[Pascal Tutorial](#)

```
In [ ]: program Lab7; // сохранить как lab7.dpr (DPR  Delphi PReject)
{$APPTYPE CONSOLE}
Uses Windows; // для русификации

const
  A: array [1..30] of integer = (1,2,3,4,5,6,7,8,9,10,
    11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30);
var
  i, j, n, m, c, MIN: integer;
  method: string;
  fin, fout: string;
  tfin, tfout: TextFile;

begin
  setConsoleCP(1251); // для ввода русификация и
  setConsoleOutputCP(1251); // для вывода

  {A0.1, A0.3 - ввод-вывод входных данных}
  writeln('Лаб.7' :40); {вывод заголовка с переходом на следующую строку по обр.4}

  {ввод-вывод строковой переменной - имени файла с входными данными по обр.1.1, 5.1}
  writeln('Текстовый файл с элементами массива A: ');
  readln(fin); writeln(fin);

  {ввод-вывод строковой переменной - имени файла с выходными данными по обр.1.2, 5.2}
  writeln('Текстовый файл с элементами массива A и элементами отсортированного массива A: ');
  readln(fout); writeln(fout);

  {ввод-вывод строковой переменной - названия метода сортировки по обр.2, 6}
  writeln('Выберете метод сортировки массива A (simple или bubble): ');
  readln(method); writeln(method);

  AssignFile(tfin, fin); reset(tfin); {открыть файл для чтения}
  AssignFile(tfout, fout); rewrite(tfout); {открыть файл для записи}

  {ввод-вывод исходного массива A по обр.3.1, 3.2, 7.1, 7.2}
  writeln('Сортировка массива A: '); {вывод пояснения с переходом на следующую строку}
```

```

i := 1;
while not eof(tfin) do
begin
    readln(tfin, A[i]); {ввод элемента и переход на следующую строку}
    writeln(tfout, A[i]:4);
    {вывод элемента исходного массива и переход на следующую строку в файле}
    i := i + 1;
end;
n := i - 1;

CloseFile(tfin); {закрыть файл для чтения}

if method = 'simple' then
begin
    for j := 0 to n-1 do
    begin
        MIN := 10000;
        for i := 1 to n-j do
        begin
            write(A[i]:4); {вывод элемента на экране}
            if A[i] < MIN then
            begin
                MIN := A[i];
                m := i;
            end;
        end;
        A[m] := A[n-j]; A[n-j] := MIN;
        writeln;
    end;
end;

if method = 'bubble' then
begin
    for j := 0 to n-1 do
    begin
        c := 0;
        write(A[1]:4);
        for i := 2 to n-j do
        begin
            write(A[i]:4); {вывод элемента на экране}
            if A[i-1] < A[i] then
            begin
                MIN := A[i-1];
                A[i-1] := A[i];
                A[i] := MIN;
                c := c + 1;
            end;
        end;
        writeln;
        if c = 0 then break;
    end;
end;

{отделим визуально чертой и строкой результат сортировки}
for i:=1 to 80 do write('=');
writeln;
for i := 1 to n do
begin
    write(A[i]:4); {вывод элемента сортированного массива на экране}
    writeln(tfout, A[i]:4);
    {вывод элемента сортированного массива и переход на следующую строку в файле}
end;

CloseFile(tfout); {закрыть файл для записи}
end.

```

Вариант программы с именами файлов, заданными в качестве параметров.

```

In [ ]: program Lab7; // сохранить как lab7.dpr (DPR H Delphi PRoject)
{$APPTYPE CONSOLE}
Uses Windows; // для русификации

const
    A: array [1..30] of integer = (1,2,3,4,5,6,7,8,9,10,
    11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30);
var
    i, j, n, m, c, MIN: integer;
    method: string;
    tfin, tfout: TextFile;

begin
    setConsoleCP(1251); // для ввода русификация и
    setConsoleOutputCP(1251); // для вывода

    {A0.1, A0.3 - ввод-вывод входных данных}
    writeln('Ла6.7' :40); {вывод заголовка с переходом на следующую строку по обр.4}

    if (ParamCount<2) then
    begin
        writeln('CritError: Не заданы параметры программы');
        readln; exit;
    end;

    AssignFile(tfin, ParamStr(1)); reset(tfin); {открыть файл для чтения}
    AssignFile(tfout, ParamStr(2)); rewrite(tfout); {открыть файл для записи}

    {ввод-вывод строковой переменной - названия метода сортировки по обр.2, 6}
    writeln('Выберете метод сортировки массива A (simple или bubble): ');
    readln(method); writeln(method);

    {ввод-вывод исходного массива A по обр.3.1, 3.2, 7.1, 7.2}
    writeln('Сортировка массива A: '); {вывод пояснения с переходом на следующую строку}

```

```

i := 1;
while not eof(tfin) do
begin
    readln(tfin, A[i]); {вывод элемента и переход на следующую строку}
    writeln(tfout, A[i]:4);
    {вывод элемента исходного массива и переход на следующую строку в файле}
    i := i + 1;
end;
n := i - 1;

CloseFile(tfin); {закрыть файл для чтения}

if method = 'simple' then
begin
    for j := 0 to n-1 do
    begin
        MIN := 10000;
        for i := 1 to n-j do
        begin
            write(A[i]:4); {вывод элемента на экране}
            if A[i] < MIN then
            begin
                MIN := A[i];
                m := i;
            end;
        end;
        A[m] := A[n-j]; A[n-j] := MIN;
        writeln;
    end;
end;

if method = 'bubble' then
begin
    for j := 0 to n-1 do
    begin
        c := 0;
        write(A[1]:4);
        for i := 2 to n-j do
        begin
            write(A[i]:4); {вывод элемента на экране}
            if A[i-1] < A[i] then
            begin
                MIN := A[i-1];
                A[i-1] := A[i];
                A[i] := MIN;
                c := c + 1;
            end;
        end;
        writeln;
        if c = 0 then break;
    end;
end;

{отделим визуально чертой и строкой результат сортировки}
for i:=1 to 80 do write('=');
writeln;
for i := 1 to n do
begin
    write(A[i]:4); {вывод элемента сортированного массива на экране}
    writeln(tfout, A[i]:4);
    {вывод элемента сортированного массива и переход на следующую строку в файле}
end;


CloseFile(tfout); {закрыть файл для записи}
end.

```

Вариант программы для онлайн-компилятора.

[Compile and Execute Pascal Online](#)

```

In [ ]: // main.pas
program Lab7; // сохранить как lab7.dpr (DPR  Delphi Project)

const
    A: array [1..30] of integer = (1,2,3,4,5,6,7,8,9,10,
    11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30);
var
    i, j, n, m, c, MIN: integer;
    method: string;

begin
    {A0.1, A0.3 - ввод-вывод входных данных}
    writeln('Лаб.7' :40); {вывод заголовка с переходом на следующую строку по обр.4}

    {ввод-вывод строковой переменной - названия метода сортировки по обр.2, 6}
    writeln('Выберете метод сортировки массива A (simple или bubble): ');
    readln(method);
    writeln(method);

    {ввод-вывод исходного массива A по обр.3.1, 3.2, 7.1, 7.2}
    writeln('Сортировка массива A: '); {вывод пояснения с переходом на следующую строку}

    i := 2;
    while not eoln do
    begin
        readln(A[i-1]); {ввод элемента и переход на следующую строку}
        i := i + 1;
    end;
    n := i - 2;

```

```

if method = 'simple' then
begin
  for j := 0 to n-1 do
  begin
    MIN := 10000;
    for i := 1 to n-j do
    begin
      write(A[i]:4); {вывод элемента на экране}
      if A[i] < MIN then
      begin
        MIN := A[i];
        m := i;
      end;
    end;
    A[m] := A[n-j]; A[n-j] := MIN;
    writeln;
  end;
end;

if method = 'bubble' then
begin
  for j := 0 to n-1 do
  begin
    c := 0;
    write(A[1]:4);
    for i := 2 to n-j do
    begin
      write(A[i]:4); {вывод элемента на экране}
      if A[i-1] < A[i] then
      begin
        MIN := A[i-1];
        A[i-1] := A[i];
        A[i] := MIN;
        c := c + 1;
      end;
    end;
    writeln;
    if c = 0 then break;
  end;
end;

{отделим визуально чертой и строкой результат сортировки}
for i:=1 to 80 do write('=');
writeln;
for i := 1 to n do write(A[i]:4);

end.

```

```

In [ ]: // STDIN
simple
-78
26
-3
32
-48
-97
-80
49
86
-32
91
58
82
19
98
-65
-93
40
6
-47

```