

Спецификация к Лабораторной работе №5 "Экстремум"

(Вариант 22)

Абстракция A0

1. Постановка задачи.

Задание:

Написать программу обработки одномерного массива в соответствии с условием.

Условие:

Для заданного массива $\{A_1, A_2, \dots, A_n\}$ определить наименьшее значение сумм $A_i + A_{i-1}$.

2. Уточненная постановка задачи.

Дан одномерный вещественный массив A из n элементов.

Найти $\min_i (A_i + A_{i-1})$, $i \in \{2, \dots, n\}$.

3. Пример с иллюстрацией.

```
In [117]: import pylab
import numpy
%matplotlib inline
pylab.style.use('seaborn-whitegrid')

# A_Lab5_1.txt, A_Lab5_2.txt, A_Lab5_3.txt, A_Lab5_4.txt, A_Lab5_5.txt
# AMIN_Lab5_1.txt, AMIN_Lab5_2.txt, AMIN_Lab5_3.txt, AMIN_Lab5_4.txt, AMIN_Lab5_5.txt

print(20*' ' + 'Лаб.5\n' + 90*'=')

fin = str(input('Текстовый файл с элементами массива A: '))
fout = str(input('Текстовый файл с элементами массива A и минимальной суммой соседних элементов: '))

with open(fin, 'r') as f:
    A = [float(x) for x in f.readlines()]
n = len(A)
MIN = 10**10
MIN_ID = 10**5

print(90*'= ' + '\n' + 'Массив A: \n', A[0])
for i in range(1, n):
    print(A[i])
    if (A[i] + A[i-1] < MIN):
        MIN = A[i] + A[i-1]
        MIN_ID = i

print(90*'= ' + '\n' + 'Минимальная сумма соседних элементов массива A:', MIN)

with open(fout, 'w') as f:
    for i in range(0, n):
        f.write("{}\n".format(A[i]))
    f.write("{}".format(MIN))

pylab.figure(figsize=(10,10))
pylab.scatter(range(n), A)
pylab.scatter([MIN_ID - 1, MIN_ID], [A[MIN_ID - 1], A[MIN_ID]],
              s=150, facecolors='none', edgecolors='r')

pylab.xlabel('i', fontsize=15)
pylab.ylabel('A[i]', fontsize=15)
pylab.xticks(range(-1, n+1), range(n+2));
```

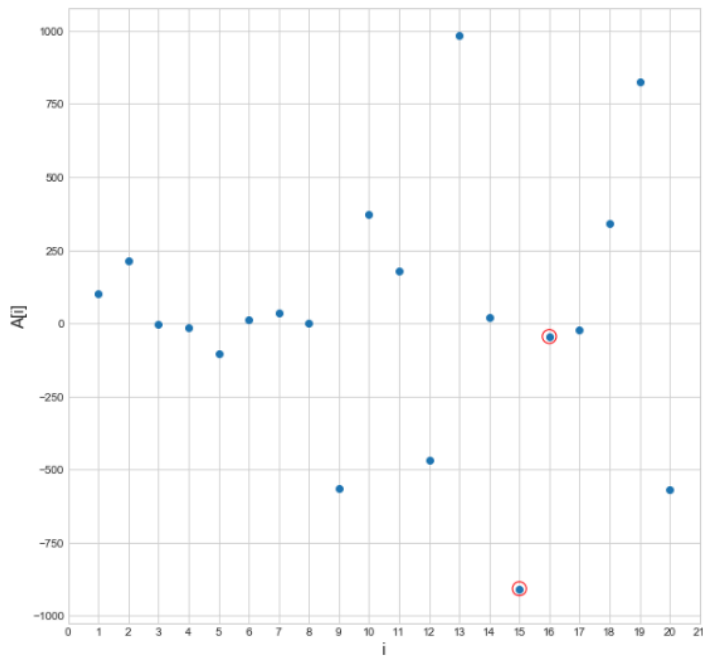
Текстовый файл с элементами массива A: A_Lab5_4.txt

Текстовый файл с элементами массива A и минимальной суммой соседних элементов: AMIN_Lab5_4.txt

Массив A:

101.5
212.6
-3.08
-15.876
-104.2
12.81
33.17
1.9
-564.22
371.4
178.23
-468.11
982.36
19.2
-908.1
-45.5
-22.345
342.6
825.33
-567.4

Минимальная сумма соседних элементов массива A: -953.6



In [118]: `print(A)`

[101.5, 212.6, -3.08, -15.876, -104.2, 12.81, 33.17, 1.9, -564.22, 371.4, 178.23, -468.11, 982.36, 19.2, -908.1, -45.5, -22.345, 342.6, 825.33, -567.4]

In [119]: `n`

Out[119]: 20

In [120]: `MIN`

Out[120]: -953.6

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	"X.txt"
входные данные	fout	название текстового файла с элементами массива A	символ	строка	"X.txt"
входные данные	A	вводимые из текстового файла числа	вещ	одномерный массив (20)	+XX.XX+ (:5:6)
выходные данные	MIN	выводимая в текстовый файл минимальная сумма соседних элементов массива	вещ	простая переменная	+XX.XX+ (:5:6)
промежуточные данные	PRE	предыдущий элемент для A[i]	вещ	простая переменная	+XX.XX+ (:5:6)
промежуточные данные	i	индекс текущего элемента, 1 ≤ i ≤ 21	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: $\langle fin \rangle$
- обр 1.2 Текстовый файл с элементами массива A и минимальной суммой соседних элементов: $\langle fout \rangle$
- обр 2.1 Массив A:
- обр 2.2 $\langle A[1] \rangle$
 $\langle A[2] \rangle$
...
 $\langle A[n] \rangle$

6. Выходная форма

- обр 3 Лаб. 5
- обр 4.1 Текстовый файл с элементами массива A: $\langle fin \rangle$
- обр 4.2 Текстовый файл с элементами массива A и минимальной суммой соседних элементов: $\langle fout \rangle$
- обр 5.1 Массив A:
- обр 5.2 $\langle A[1] \rangle$
 $\langle A[2] \rangle$
...
 $\langle A[n] \rangle$
- обр 6 Минимальная сумма соседних элементов массива A: $\langle MIN \rangle$

7. Аномалии

-

8. Функциональные тесты

Исходные данные	-	-	Результаты	Тест
fin	fout	A	MIN	№
A_Lab5_1.txt	AMIN_Lab5_1.txt	[11, 20, -35, 100, -56, -71, -89, 90, 33, 25, 16, 67, 112, -107, 64, 38, 22, 45, -93]	-160	1
-	-	-	-	-
A_Lab5_2.txt	AMIN_Lab5_2.txt	[0, 0, 0, -1, -1, 0, 0, 0]	-2	2
-	-	-	-	-
A_Lab5_3.txt	AMIN_Lab5_3.txt	[10, 8, 6, 4, 2, 0, -1, -3, -5, -7, -9]	-16	3
-	-	-	-	-
A_Lab5_4.txt	AMIN_Lab5_4.txt	[101.5, 212.6, -3.08, -15.876, -104.2, 12.81, 33.17, 1.9, -564.22, 371.4, 178.23, -468.11, 982.36, 19.2, -908.1, -45.5, -22.345, 342.6, 825.33, -567.4]	-953.6	4
-	-	-	-	-
-	-	-	-	-
A_Lab5_5.txt	AMIN_Lab5_5.txt	[1000.0, 2000.0, 3000.0, 4000.0, 5000.0, -5000.0, -4000.0, -3000.0, -2000.0, -1000.0]	-9000.0	5

№ теста	Входные данные	-	-	Ожидаемый результат	Смысл теста
-	-	-	-	-	-
1	fin = 'A_Lab5_1.txt'	fout = 'AMIN_Lab5_1.txt'	A = [11, 20, -35, 100, -56, -71, -89, 90, 33, 25, 16, 67, 112, -107, 64, 38, 22, 45, -93]	MIN = -160	Подтвердить правильность расчетов в случае целочисленного массива
-	-	-	-	-	-
2	fin = 'A_Lab5_2.txt'	fout = 'AMIN_Lab5_2.txt'	A = [0, 0, 0, -1, -1, 0, 0, 0]	MIN = -2	Протестировать простейший случай
-	-	-	-	-	-
3	fin = 'A_Lab5_3.txt'	fout = 'AMIN_Lab5_3.txt'	A = [10, 8, 6, 4, 2, 0, -1, -3, -5, -7, -9]	MIN = -16	Протестировать убывающую последовательность
-	-	-	-	-	-
4	fin = 'A_Lab5_4.txt'	fout = 'AMIN_Lab5_4.txt'	A = [101.5, 212.6, -3.08, -15.876, -104.2, 12.81, 33.17, 1.9, -564.22, 371.4, 178.23, -468.11, 982.36, 19.2, -908.1, -45.5, -22.345, 342.6, 825.33, -567.4]	MIN = -953.6	Подтвердить правильность расчетов в случае вещественного массива
-	-	-	-	-	-
-	-	-	-	-	-
5	fin = 'A_Lab5_5.txt'	fout = 'AMIN_Lab5_5.txt'	A = [1000.0, 2000.0, 3000.0, 4000.0, 5000.0, -5000.0, -4000.0, -3000.0, -2000.0, -1000.0]	MIN = -9000.0	Протестировать последовательность, состоящую из двух возрастающих

-	Результаты	№ теста
-	-	-
MIN	Максимальная вычислительная нагрузка – 18 сумм соседних элементов.	1
-	Соответствует ожидаемым числовым значениям.	-
MIN	-2	2
MIN	Максимальная вычислительная нагрузка – 7 сумм соседних элементов.	3
-	Равно сумме последней пары.	-
MIN	Максимальная вычислительная нагрузка – 19 сумм соседних элементов.	4
-	Соответствует ожидаемым числовым значениям.	-
MIN	Максимальная вычислительная нагрузка – 9 сумм соседних элементов.	5
-	Соответствует ожидаемым числовым значениям.	-

9. Метод

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

1. *Подзадача А 0.1.* Запросить названия текстовых файлов ввода-вывода, считать из текстового файла ввода исходные данные (обр.1-2), затем вывести их на экран (обр.3-5) для визуального подтверждения.
2. *Подзадача А 0.2.* Решение поставленной задачи: на основе введенных исходных данных (A) подсчитать минимальную сумму соседних элементов этого массива.
3. *Подзадача А 0.3.* Ввести полученные результаты (MIN) на экран (обр.6) и записать в файл вывода исходный массив и полученный результат.

Ввод-вывод тривиален и представляет собой чисто техническую задачу.

Для осуществления процесса ввода-вывода массива потребуются промежуточные переменные i – индекс текущего элемента массива и предыдущий элемент PRE для значения $A[i]$, их следует добавить в таблицу данных.

Алгоритм необходим только для обработки входных данных и расчета результата.

10. Алгоритм

[Sample diagrams](#)

```
In [85]: %%file block_diagram
```

```
blockdiag {
    orientation = portrait;
    class start_end [shape = ellipse, fontsize = 15, color = lightskyblue];
    class input_out [shape = flowchart.input, fontsize = 18, width = 650, color = lightcyan];
    class main_box [shape = box, width = 500, height = 150, fontsize = 18];

    Начало ->
    "Вывод заголовка по обр.3 на экран" ->
    "Ввод имени файла по обр.1.1" -> "Ввод имени файла по обр.1.2" ->
    "Ввод пояснения по обр.2.1" -> "Ввод A из файла по обр.2.2" ->
    "Вывод имени файла по обр.4.1 на экран" -> "Вывод имени файла по обр.4.2 на экран" ->
    "Вывод пояснения по обр.5.1 на экран" -> "Вывод A на экран и в файл по обр.5.2" ->
    "Подсчет минимальной суммы \псоседних элементов массива A \n(MIN)" ->
    "Вывод MIN на экран и в файл по обр.6" ->
    Конец;

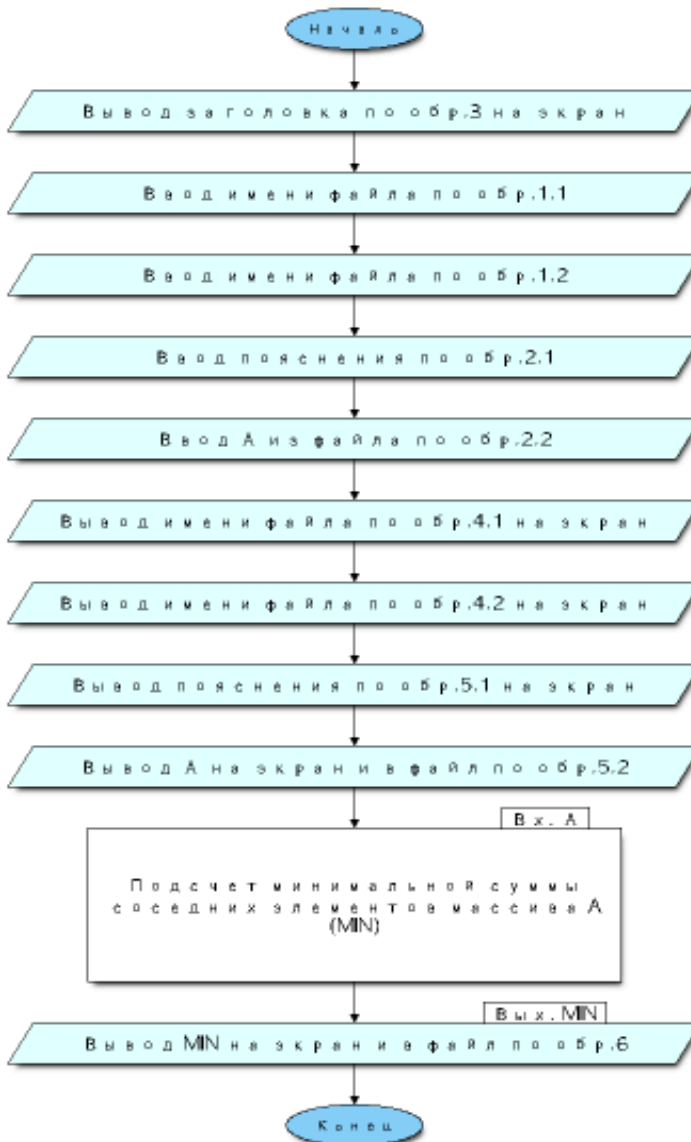
    Начало, Конец [class = "start_end"];
    "Ввод имени файла по обр.1.1", "Ввод имени файла по обр.1.2" [class = "input_out"];
    "Ввод пояснения по обр.2.1", "Ввод A из файла по обр.2.2" [class = "input_out"];
    "Вывод имени файла по обр.4.1 на экран", "Вывод имени файла по обр.4.2 на экран" [class = "input_out"];
    "Вывод пояснения по обр.5.1 на экран", "Вывод A на экран и в файл по обр.5.2" [class = "input_out"];
    "Вывод заголовка по обр.3 на экран", "Вывод MIN на экран и в файл по обр.6" [class = "input_out"];
    "Подсчет минимальной суммы \псоседних элементов массива A \n(MIN)" [class = "main_box"];

    "Вывод A на экран и в файл по обр.5.2" ->
    "Подсчет минимальной суммы \псоседних элементов массива A \n(MIN)" [label = 'Bx. A', fontsize = 18];
    "Подсчет минимальной суммы \псоседних элементов массива A \n(MIN)" ->
    "Вывод MIN на экран и в файл по обр.6" [label = 'Вых. MIN', fontsize = 18];
}
```

Overwriting block_diagram

```
In [86]: !blockdiag block_diagram
```

```
In [87]: from IPython.display import Image
Image("block_diagram.png")
```



11. Программа на *Delphi*.

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

Синтаксис:

[Pascal Tutorial](#)

Проверить программу онлайн без установки программной среды:

[Compile and Execute Pascal Online](#)

```
In [ ]: program Lab5; //сохранить как lab5.dpr (DPR  Delphi Project)
{$APPTYPE CONSOLE}
Uses Windows; // для русификации (сделать комментарием в онлайн-компиляторе!)

var
  i: integer;
  PRE, MIN: real;
  fin, fout: string;
  tfin, tfout: TextFile;
  A: array of real;

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('Лаб6.5' :40); {вывод заголовка с переходом на следующую строку по обр.3}

  {ввод-вывод строковой переменной - имени файла с входными данными по обр.1.1, 4.1}
  writeln('Текстовый файл с элементами массива A: ');
  readln(fin);
  writeln(fin);

  {ввод-вывод строковой переменной - имени файла с входными данными по обр.1.2, 4.2}
  writeln('Текстовый файл с элементами массива A и минимальной суммой соседних элементов: ');
  readln(fout);
  writeln(fout);

  {ввод-вывод исходного массива A по обр.2.1, 2.2, 5.1, 5.2}
  writeln('Массив 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;
  MIN := 100000000000;

  while not eof(tfin) do

  begin
    SetLength(A,i);
    readln(tfin, A[i]); {ввод элемента и переход на следующую строку}
    writeln(A[i]:5:6); {вывод элемента и переход на следующую строку на экране}
    writeln(tfout, A[i]:5:6); {вывод элемента и переход на следующую строку в файле}

    MIN := -160; {заглушка для теста 1}
    // MIN := -2; {заглушка для теста 2}
    // MIN := -16; {заглушка для теста 3}
    // MIN := -953.6; {заглушка для теста 4}
    // MIN := -9000.0; {заглушка для теста 5}

    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;

    writeln('Минимальная сумма соседних элементов массива A:', MIN:5:6);
    {вывод переменной и переход на следующую строку на экране по обр.6}
    writeln(tfout, MIN:5:6);
    {вывод MIN и переход на следующую строку в файле по обр.6}

    CloseFile(tfout); {закрыть файл для записи}

  end.
```

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 текущей точки,
- $A[i]$ – соответствующий элемент исходного массива A ,
- PRE – предыдущий элемент для $A[i]$,
- MIN – минимальная сумма соседних элементов массива A .

Присвоим переменной MIN начальное значение: $MIN = 10000000000$.

Поиск минимальной суммы осуществляется с помощью цикла.

На каждом этапе сумма соседних элементов сравнивается с текущим значением переменной MIN .

Для каждого элемента, начиная со второго, предыдущий элемент сохраняется в качестве вспомогательной переменной PRE .

Если $A[i] + PRE < MIN$, то значение переменной MIN заменяется: $MIN = A[i] + PRE$.

10. Алгоритм

```
In [124]: %%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 = 300, color = lightsteelblue];
    class loop_out [shape = flowchart.loopout, fontsize = 18, width = 300, color = lightsteelblue];
    class condition [shape = flowchart.condition, fontsize = 18, width = 400, color = lightsteelblue];
    class command_box [shape = roundedbox, fontsize = 18, width = 300];

    Начало -> "MIN := 10000000000" -> "i := 2; +1; n" ->
    "A[i] + PRE < MIN" -> "MIN := A[i] + PRE" ->
    "PRE = A[i]" -> "i = n" -> Конц;
    "A[i] + PRE < MIN" -> "PRE = A[i]" -> "i = n" -> Конц;

    Начало, Конц [class = "start_end"];
    "i := 2; +1; n" [class = "loop_in"];
    "i = n" [class = "loop_out"];
    "A[i] + PRE < MIN" [class = "condition"];
    "MIN := 10000000000", "MIN := A[i] + PRE", "PRE = A[i]" [class = "command_box"];

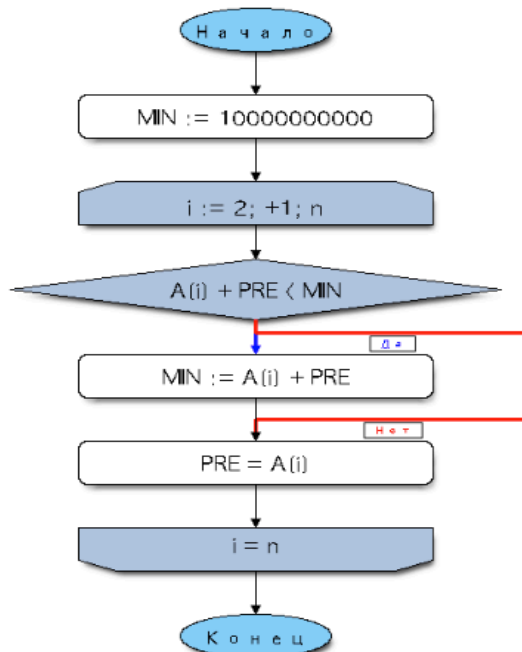
    "A[i] + PRE < MIN" -> "MIN := A[i] + PRE" [class = 'yes'];
    "A[i] + PRE < MIN" -> "PRE = A[i]" [class = 'no'];
}

Overwriting block_diagram2
```

```
In [125]: !blockdiag block_diagram2
```

```
In [126]: Image("block_diagram2.png")
```

Out[126]:




11. Программный код

```
In [ ]: MIN := 10000000000;  
  
if i > 2 then  
begin  
  if (A[i] + PRE < MIN) then MIN := A[i] + PRE;  
  PRE := A[i];  
end;
```

Получившаяся программа

```
In [ ]: program Lab5; //сохранить как lab5.dpr (DPR  Delphi Project)  
{ $APPTYPE CONSOLE}  
Uses Windows; // для русификации (сделать комментарием в онлайн-компиляторе!)  
  
var  
  i: integer;  
  PRE, MIN: real;  
  fin, fout: string;  
  tfin, tfout: TextFile;  
  A: array of real;  
  
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, A0.2, A0.3 – ввод-вывод входных данных}  
  writeln('Лаб.5' :40); {вывод заголовка с переходом на следующую строку по обр.3}  
  
  {ввод-вывод строковой переменной – имени файла с входными данными по обр.1.1, 4.1}  
  writeln('Текстовый файл с элементами массива A: ');  
  readln(fin);  
  writeln(fin);  
  
  {ввод-вывод строковой переменной – имени файла с входными данными по обр.1.2, 4.2}  
  writeln('Текстовый файл с элементами массива A и минимальной суммой соседних элементов: ');  
  readln(fout);  
  writeln(fout);  
  
  {ввод-вывод исходного массива A по обр.2.1, 2.2, 5.1, 5.2}  
  writeln('Массив 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;  
  MIN := 10000000000;  
  
  while not eof(tfin) do  
  
  begin  
    SetLength(A,i);  
    readln(tfin, A[i]); {ввод элемента и переход на следующую строку}  
    writeln(A[i]:5:6); {вывод элемента и переход на следующую строку на экране}  
    writeln(tfout, A[i]:5:6); {вывод элемента и переход на следующую строку в файле}  
  
    if i > 1 then  
    begin  
      if (A[i] + PRE < MIN) then MIN := A[i] + PRE;  
      PRE := A[i];  
    end;  
  
    i := i + 1;  
  end;  
  
  CloseFile(tfin); {закреть файл для чтения}  
  
  for i:=1 to 80 do  
    write(' '); {отделим визуально чертой и строкой введенные и выводимые значения}  
  writeln;  
  
  writeln('Минимальная сумма соседних элементов массива A:', MIN:5:6);  
  {вывод переменной и переход на следующую строку на экране по обр.6}  
  writeln(tfout, MIN:5:6);  
  {вывод MIN и переход на следующую строку в файле по обр.6}  
  
  CloseFile(tfout); {закреть файл для записи}  
  
end.
```

Та же программа с именами файлов, заданными в качестве параметров.

```
In [ ]: program Lab5; //сохранить как lab5.dpr (DPR  Delphi Project)  
{ $APPTYPE CONSOLE}  
Uses Windows; // для русификации (сделать комментарием в онлайн-компиляторе!)  
  
var  
  i: integer;  
  PRE, MIN: real;  
  fin, fout: string;  
  tfin, tfout: TextFile;  
  A: array of real;
```



```

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

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

    {A0.1, A0.2, A0.3 – ввод-вывод входных данных}
    writeln('Лаб.5' :40); {вывод заголовка с переходом на следующую строку по обр.3}

    {ввод-вывод исходного массива A по обр.2.1, 2.2, 5.1, 5.2}
    writeln('Массив A: '); {вывод пояснения с переходом на следующую строку}

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

    i := 1;
    MIN := 10000000000;

    while not eof(tfin) do
      begin
        SetLength(A,i);
        readln(tfin, A[i]); {ввод элемента и переход на следующую строку}
        writeln(A[i]:5:6); {вывод элемента и переход на следующую строку на экране}
        writeln(tfout, A[i]:5:6); {вывод элемента и переход на следующую строку в файле}

        if i > 1 then
          begin
            if (A[i] + PRE < MIN) then MIN := A[i] + PRE;
            PRE := A[i];
          end;

          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;

          writeln('Минимальная сумма соседних элементов массива A:', MIN:5:6);
          {вывод переменной и переход на следующую строку на экране по обр.6}
          writeln(tfout, MIN:5:6);
          {вывод MIN и переход на следующую строку в файле по обр.6}

          CloseFile(tfout); {закрыть файл для записи}

        end.

```