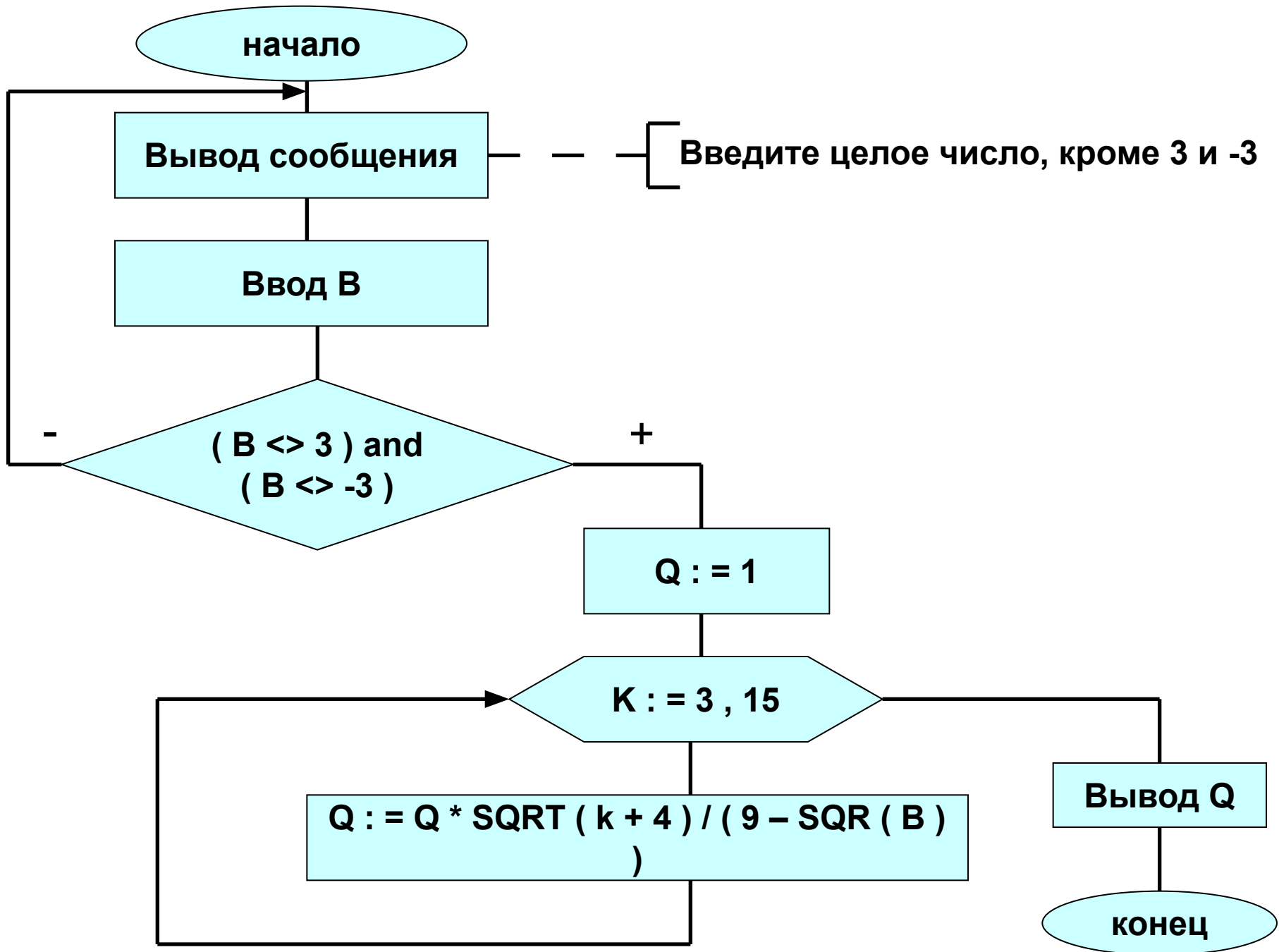


Задание:

Разработать алгоритм в виде блок-схемы и написать текст программы, для нахождения результата произведения:

$$Q = \prod_{k=3}^{15} \frac{\sqrt{k+4}}{9-b^2}$$



Program Primer ;

Uses CRT ;

Var B : integer ;

k : byte ;

Q : real ;

BEGIN

Repeat *{ начало цикла }*

ClrScr ;

Write (' Введите целое число, кроме 3 и -3 : ') ;

ReadLN (B) ;

Until (B <> 3) and (B <> -3) ; *{ критерий выхода из цикла }*

Q := 1 ;

For K := 3 **to** 15 **do**

Q := Q * SQRT (k + 4) / (9 – SQR (B)) ;

WriteLN (' Q = ' , Q : 10 : 2) ; *{ форматный вывод }*

ReadKey ; *{ задержка выполнения программы }*

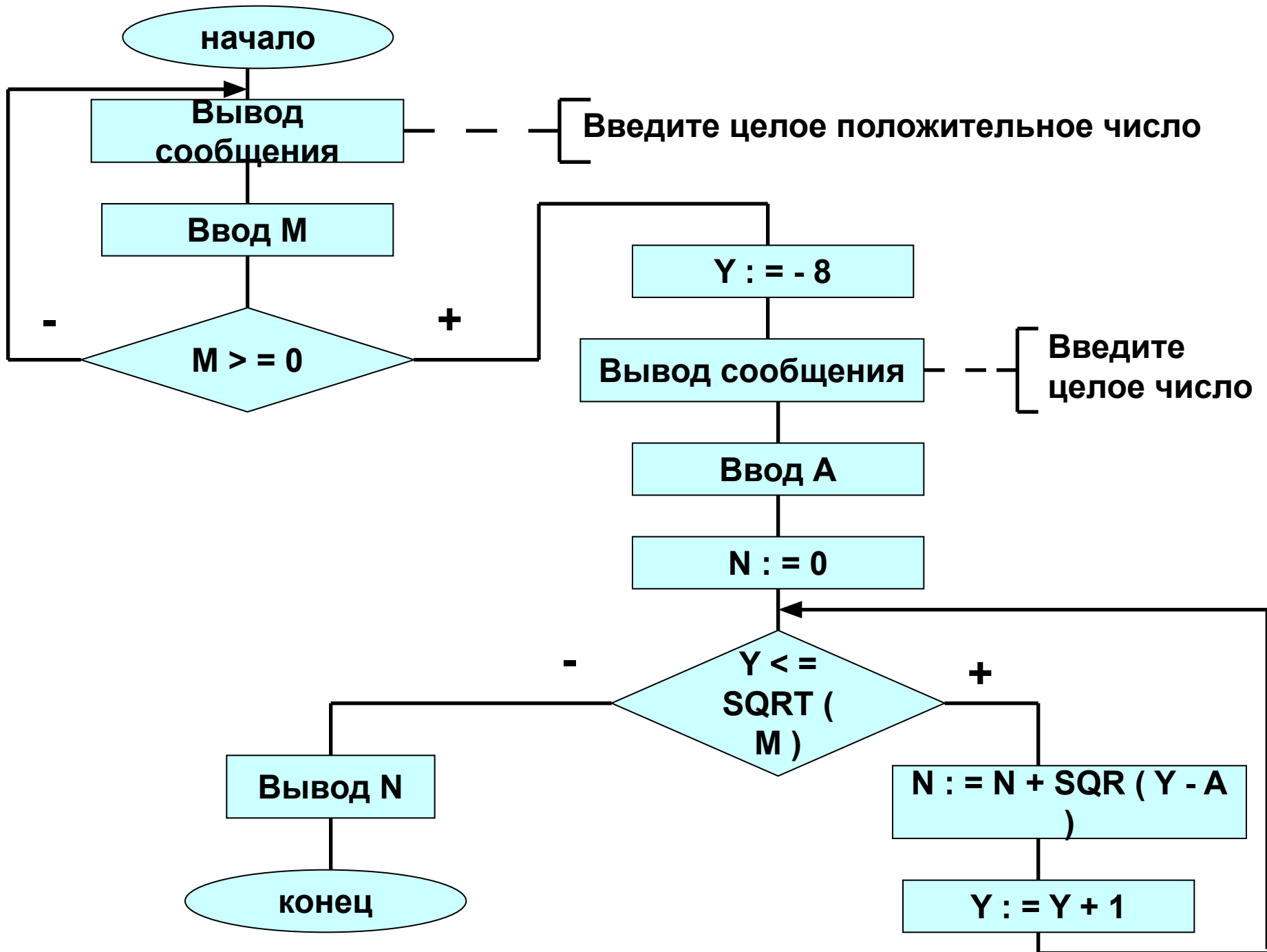
END.

Задание:

Разработать алгоритм в виде блок-схемы и написать текст программы, для нахождения суммы:

$$N = \sum_{y=-8}^{\infty} (y-a)^2$$

суммирование производить при условии, если $y \leq \sqrt{m}$



Program Primer ;

Var Y , A , N : integer ;

M : word ;

BEGIN

Repeat { начало цикла }

Write (' Введите целое положительное число : ') ;

ReadLN (M) ;

Until M >= 0 ; { критерий выхода из цикла }

Y := - 8 ; N := 0 ;

Write (' Введите целое число : ') ;

ReadLN (A) ;

While Y <= SQRT (M) do { начало цикла }

begin

N := N + SQR (Y - A) ;

Y := Y + 1 ;

end ;

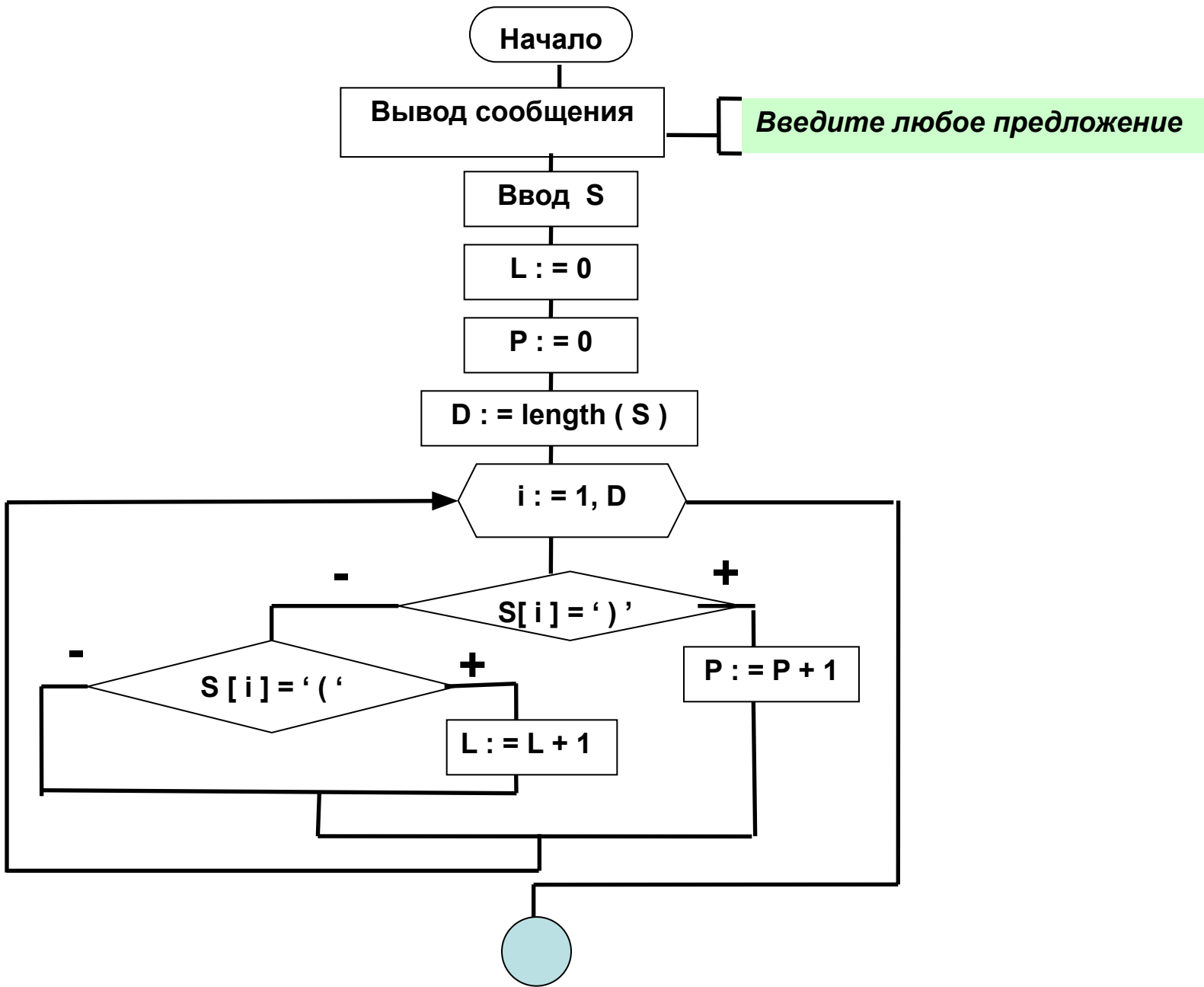
WriteLN (' N = ' , N) ;

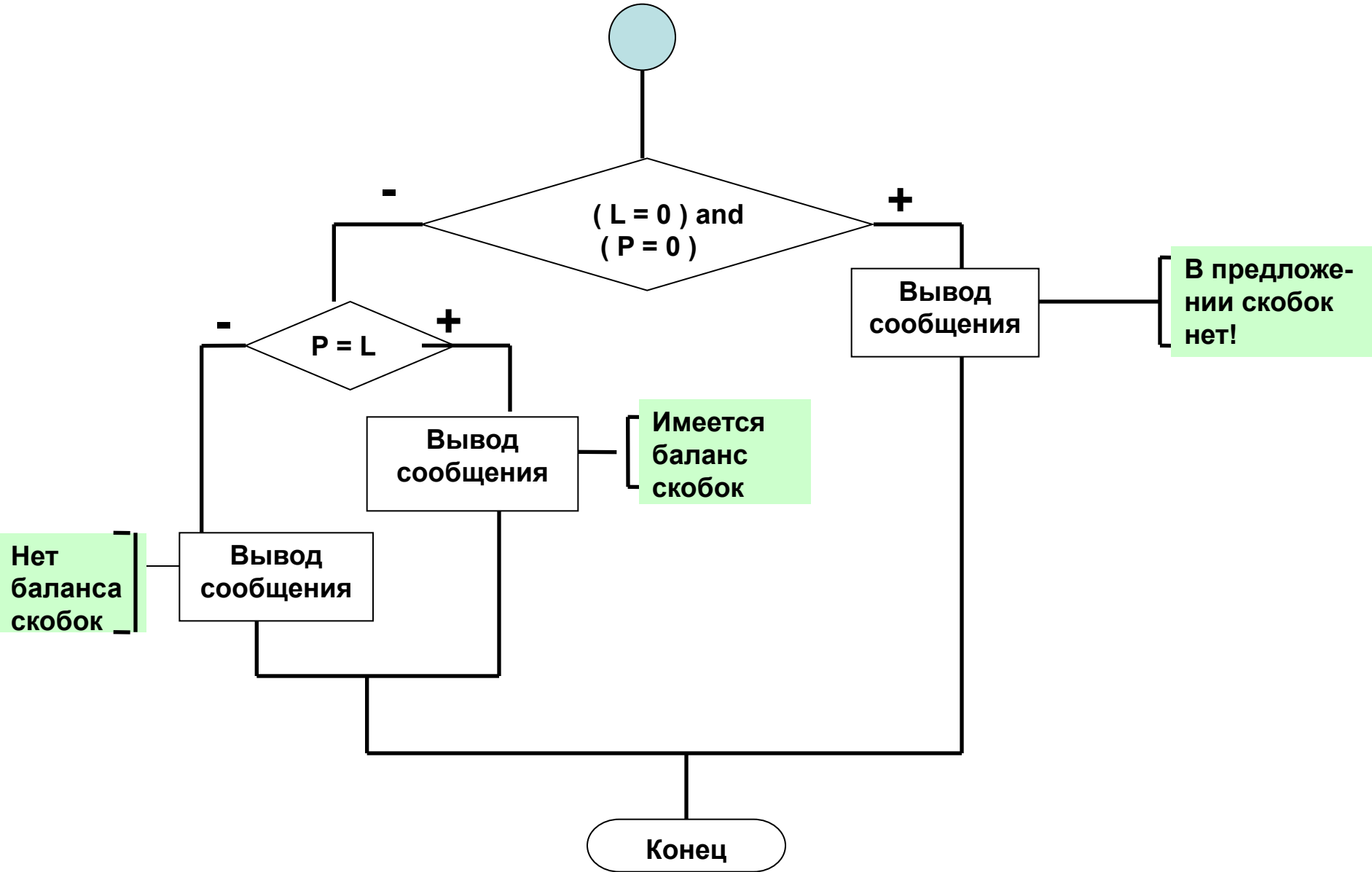
ReadLn ; { задержка выполнения программы }

END.

Пример:

Установить, имеется ли в введенном тексте баланс открывающихся и закрывающихся скобок.





Program Stroka1;

Uses CRT ; { *подключение модуля для управления экраном* }

Var S : string ; { *исходная строка* }

P : byte ; { *количество правых скобок в строке S* }

L : byte ; { *количество левых скобок в строке S* }

i : byte ; { *индексы элементов строки S* }

D : byte ; { *длина введённой строки* }

Begin

CLrScr ; { *очистка экрана* }

WriteLn (' Введите любое предложение: ') ;

ReadLn (S) ;

L := 0 ;

P := 0 ;

D := Length (S) ; { определили длину введённой строки }

For i := 1 to D do

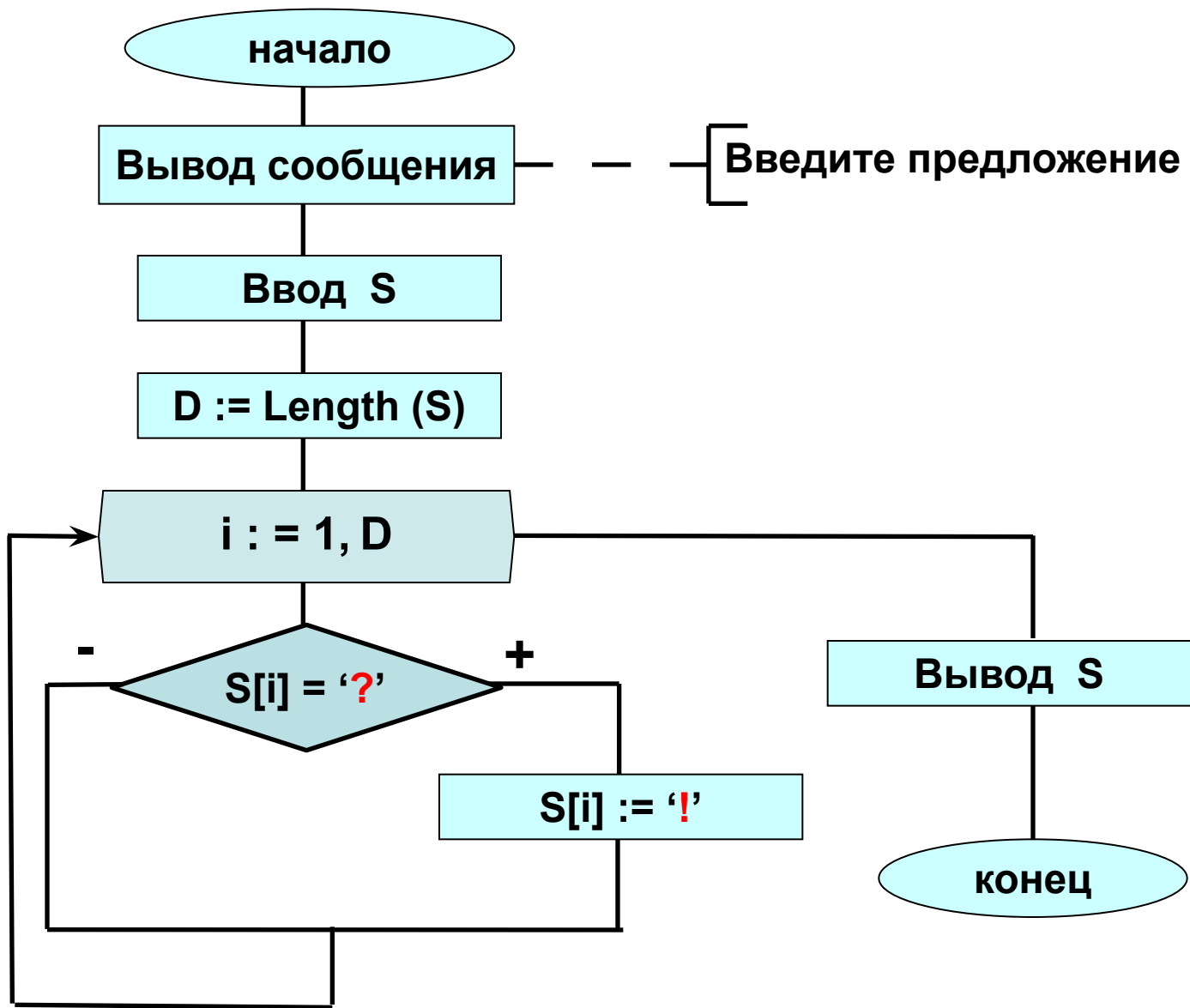
iF S[i] = ') ' then p := p + 1 eLse

iF S[i] = ' (' then L := L + 1 ;

```
IF ( L = 0 ) and ( P = 0 ) THEN  
  WriteLn ( ' В ведённом предложении скобок нет! ' )  
ELSE  
IF L = P THEN  
  WriteLn ( ' В введённой строке имеется баланс скобок ' )  
  ELSE  
    WriteLn ( ' В введённой строке нет баланса скобок ' ) ;  
ReadKey ;      { задержка выполнения программы }  
End.
```

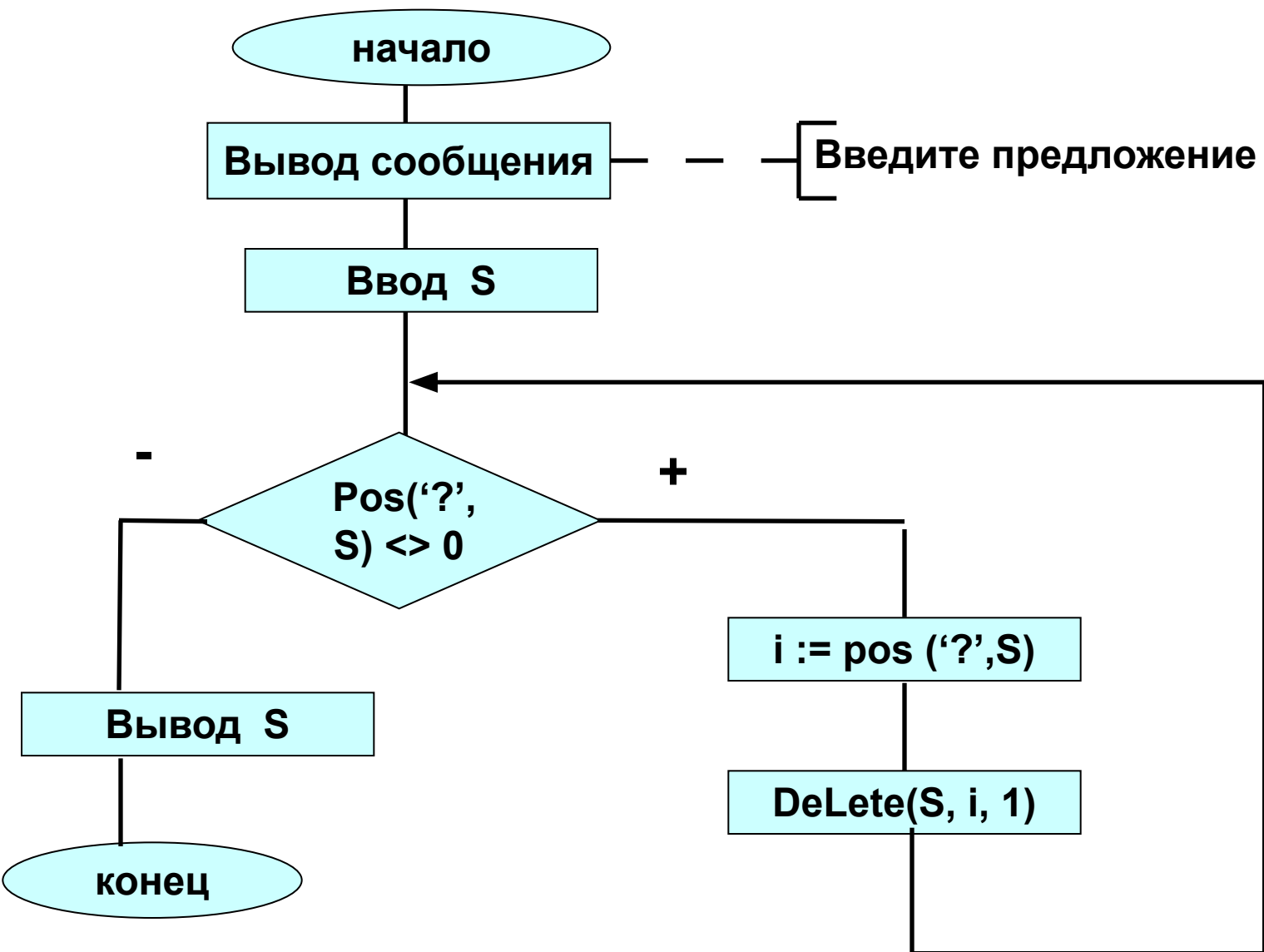
Задача:

Заменить в введённом тексте
все вопросительные знаки
на восклицательные



Задача:

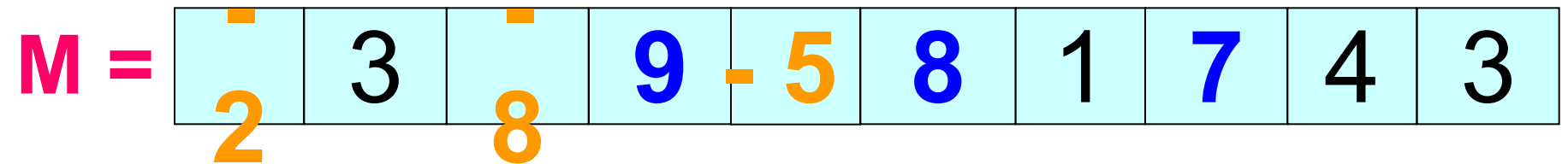
Удалить в введённом тексте
все вопросительные знаки



Задача:

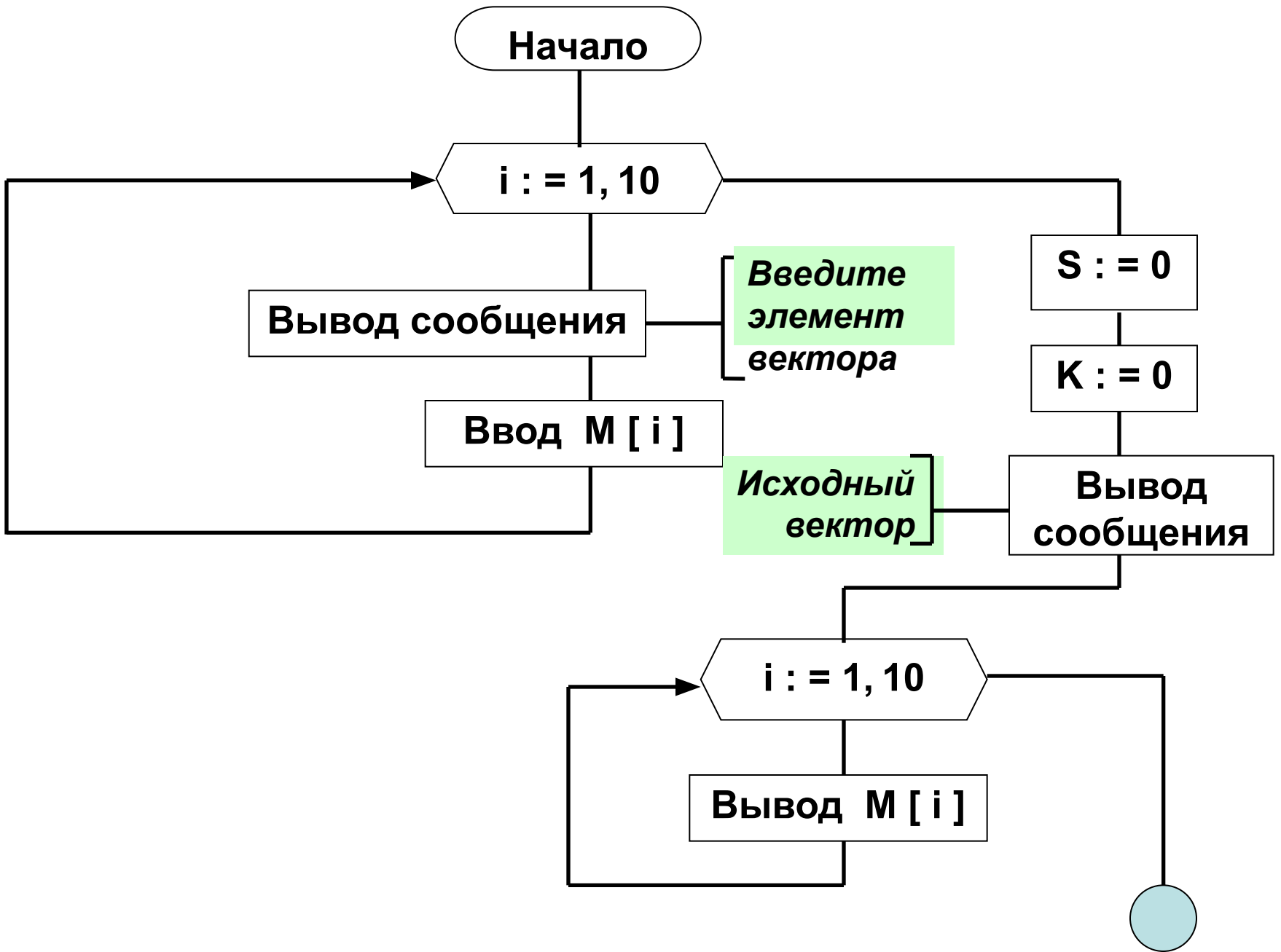
Организовать ручное заполнение целочисленного вектора M , размерностью 10.

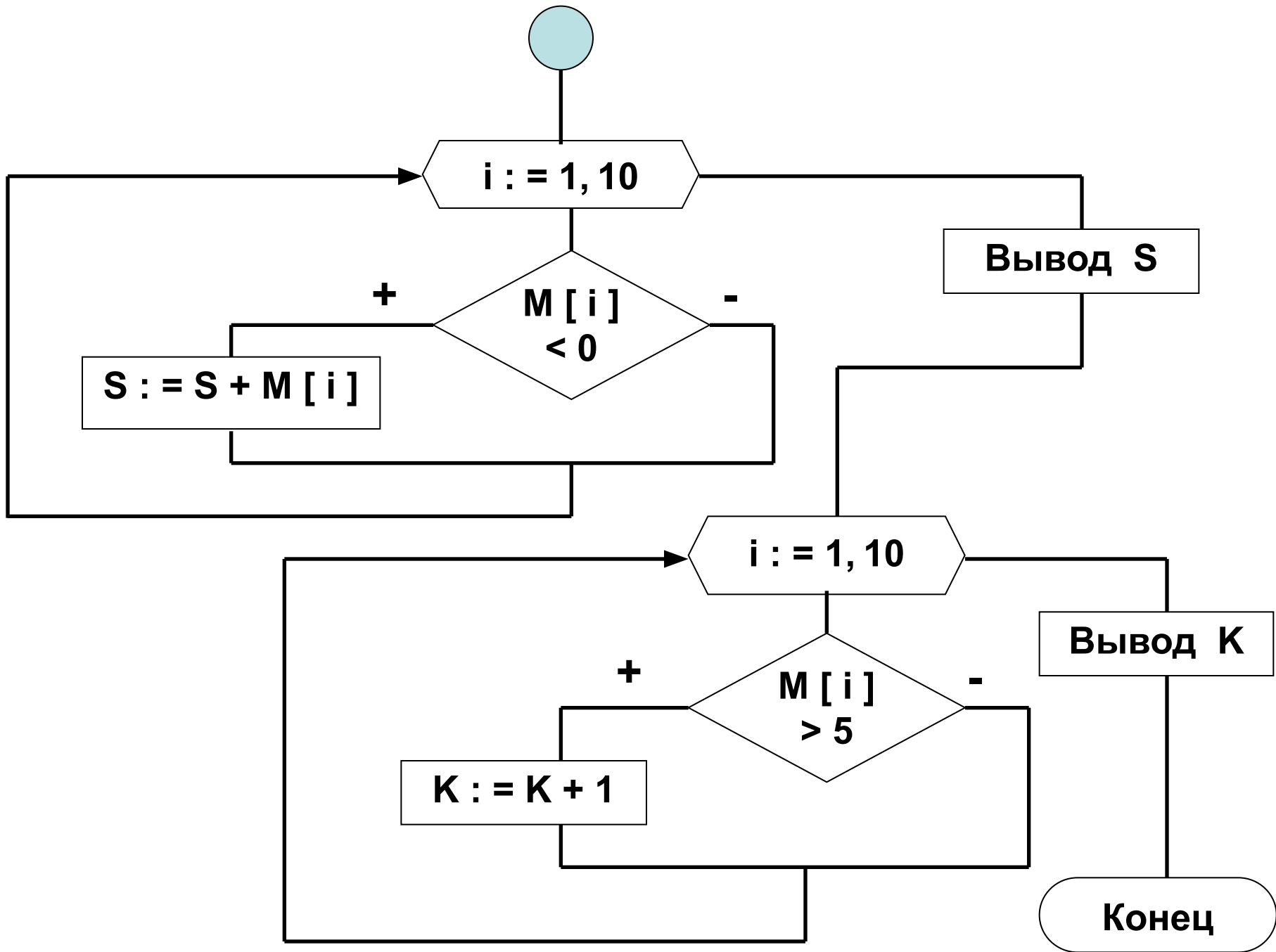
Вывести на экран элементы вектора, а также **сумму** всех отрицательных его элементов и **количество** элементов, больших числа 5.



Сумма отрицательных элементов равна - **15**

Количество элементов больших 5, равно **3**





Program Vektor ;

Var

M : array [1 .. 10] of integer ; { вектор }

i : byte ; { индекс элементов вектора }

K : byte ; { количество элементов, больших числа 5 }

S : integer ; { сумма отрицательных элементов }

BEGIN

FOR i := 1 **to** 10 **do** { заполнение вектора }

begin

Write (' Введите элемент вектора : ') ;

ReadLn (M [i]) ;

end ;

S := 0 ; { первоначальное значение суммы }

K := 0 ; { первоначальное значение количества }

Writeln (' Исходный вектор ') ;

FOR i := 1 **to** 10 **do** { вывод вектора на экран }

Write (M [i] : 7) ; { форматный вывод }

WriteLn ; { переход на новую строку }

```
FOR i := 1 to 10 do      { поиск суммы }  
IF M[i] < 0 THEN S := S + M[i];  
WriteLn ( ' Сумма отрицательных элементов вектора равна ' , S ) ;
```

```
WriteLn ; { переход на новую строку }
```

```
FOR i := 1 to 10 do      { поиск количества }  
IF M[i] > 5 THEN K := K + 1 ;  
WriteLn ( ' Количество элементов удовлетворяющих условию = ' , K ) ;
```

```
ReadLn; { задержка выполнения программы }  
END.
```

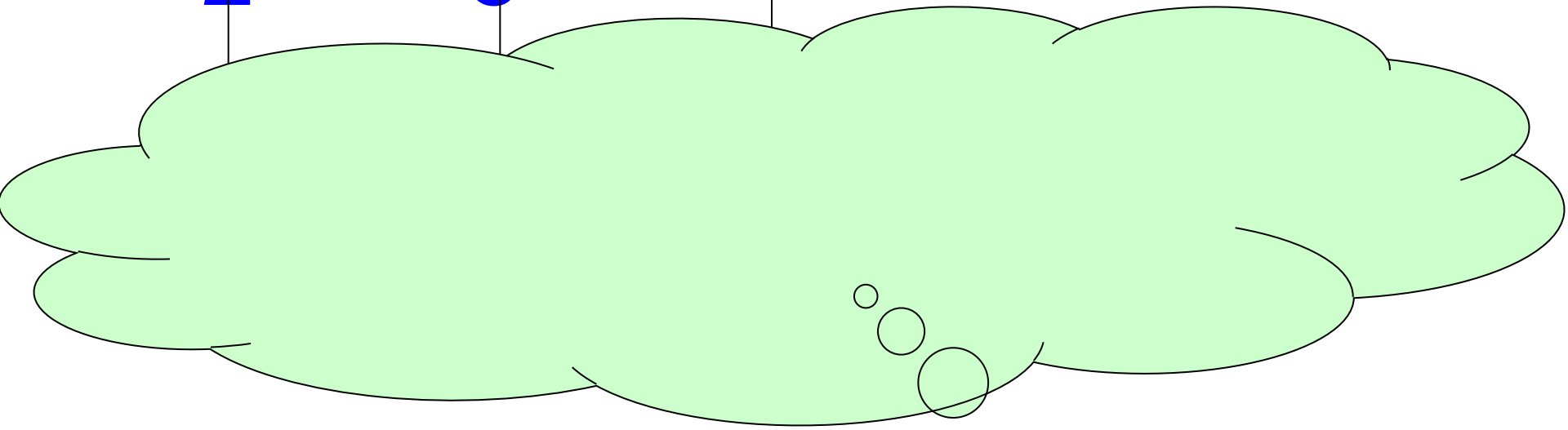
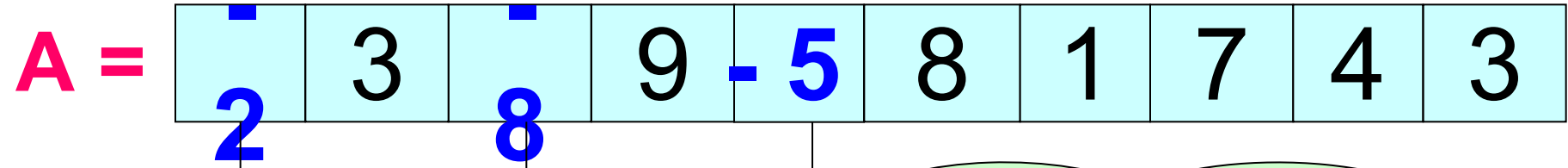
Задача:

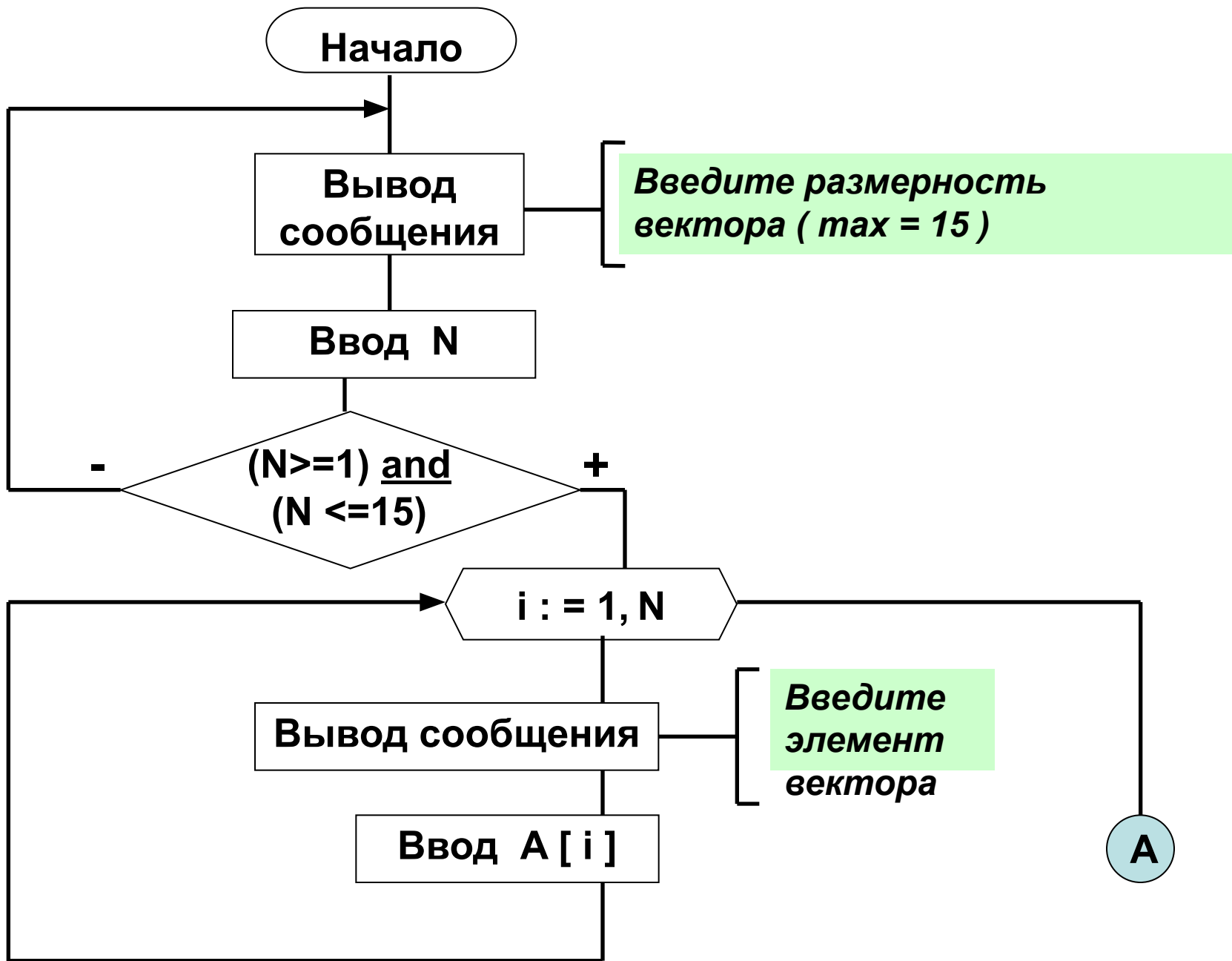
Организовать ручное заполнение целочисленного одномерного массива A , размерностью N .

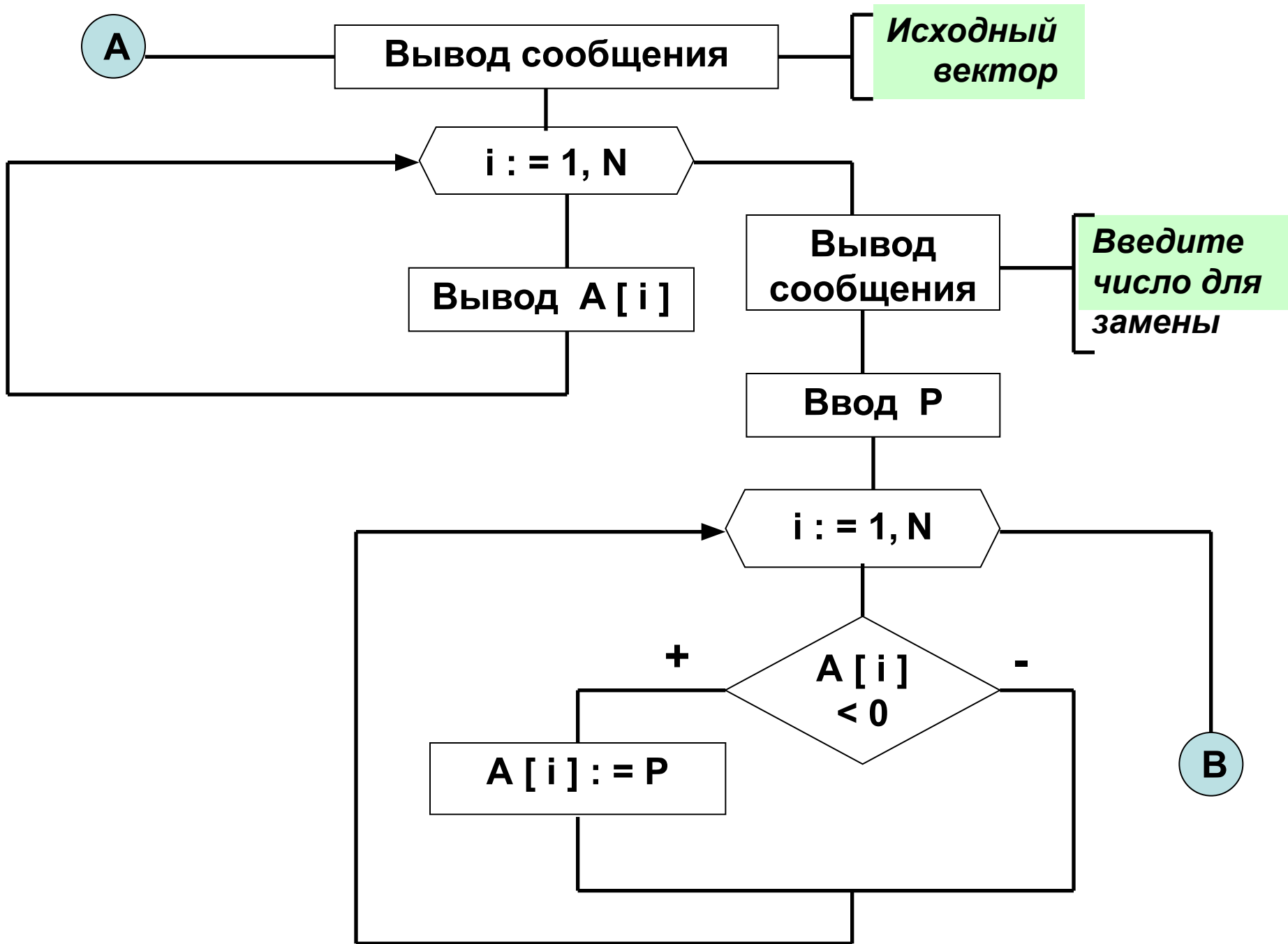
Заменить все отрицательные элементы на число P .

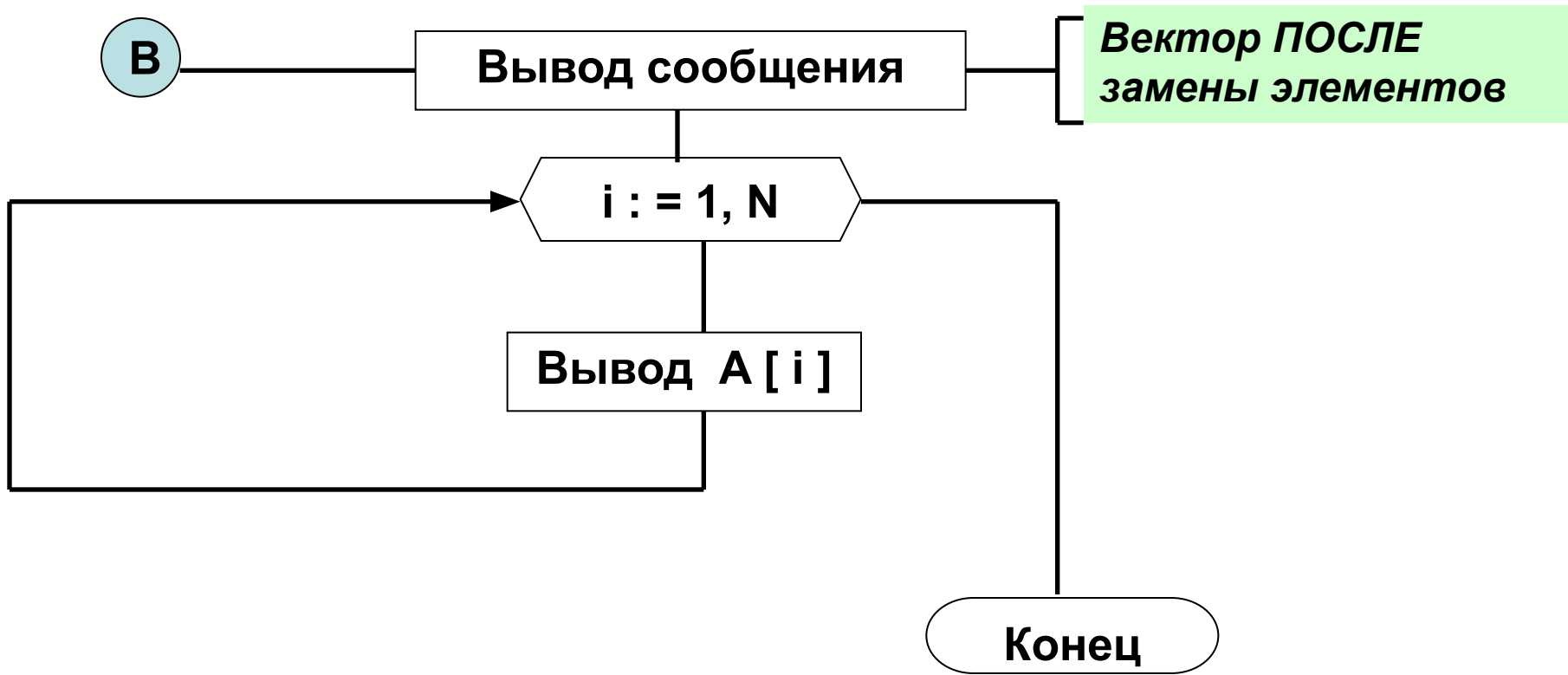
Вывести на экран элементы вектора ДО и ПОСЛЕ изменения.

$$P = 44$$









Program Vektor ;

Var

A : array [1 .. 15] of integer ; { вектор }
i : byte ; { индекс элементов вектора }
N : byte ; { количество элементов в векторе }
P : integer ; { число для замены }

BEGIN

REPEAT { проверка корректности ввода размерности }

Write (' Введите размерность вектора : ') ;

ReadLn (N) ;

UNTIL (N >= 1) and (N <= 15) ;

FOR i := 1 to N do { заполнение вектора }

begin

Write (' Введите элемент вектора : ') ;

ReadLn (A [i]) ;

end ;

Writeln (' Исходный вектор ') ;

FOR i := 1 to N do { вывод вектора на экран }

Write (A [i] : 5) ; { форматный вывод }

WriteLn ; { переход на новую строку }

Write (' Введите число для замены : ') ;
ReadLn (P) ;

FOR i := 1 **to** N **do** { замена отрицательных элементов }
IF A[i] < 0 **THEN** A[i] := P ;

WriteLn (' Вектор ПОСЛЕ замены элементов ') ;
FOR i := 1 **to** N **do** { вывод вектора на экран }
Write (A [i] : 5) ; { форматный вывод }

ReadLn; { задержка выполнения программы }
END.

Задача:

Организовать заполнение целочисленного вектора Y , размерностью 10, **случайным образом**.

Найти в массиве **минимальный** элемент и вывести его на экран.

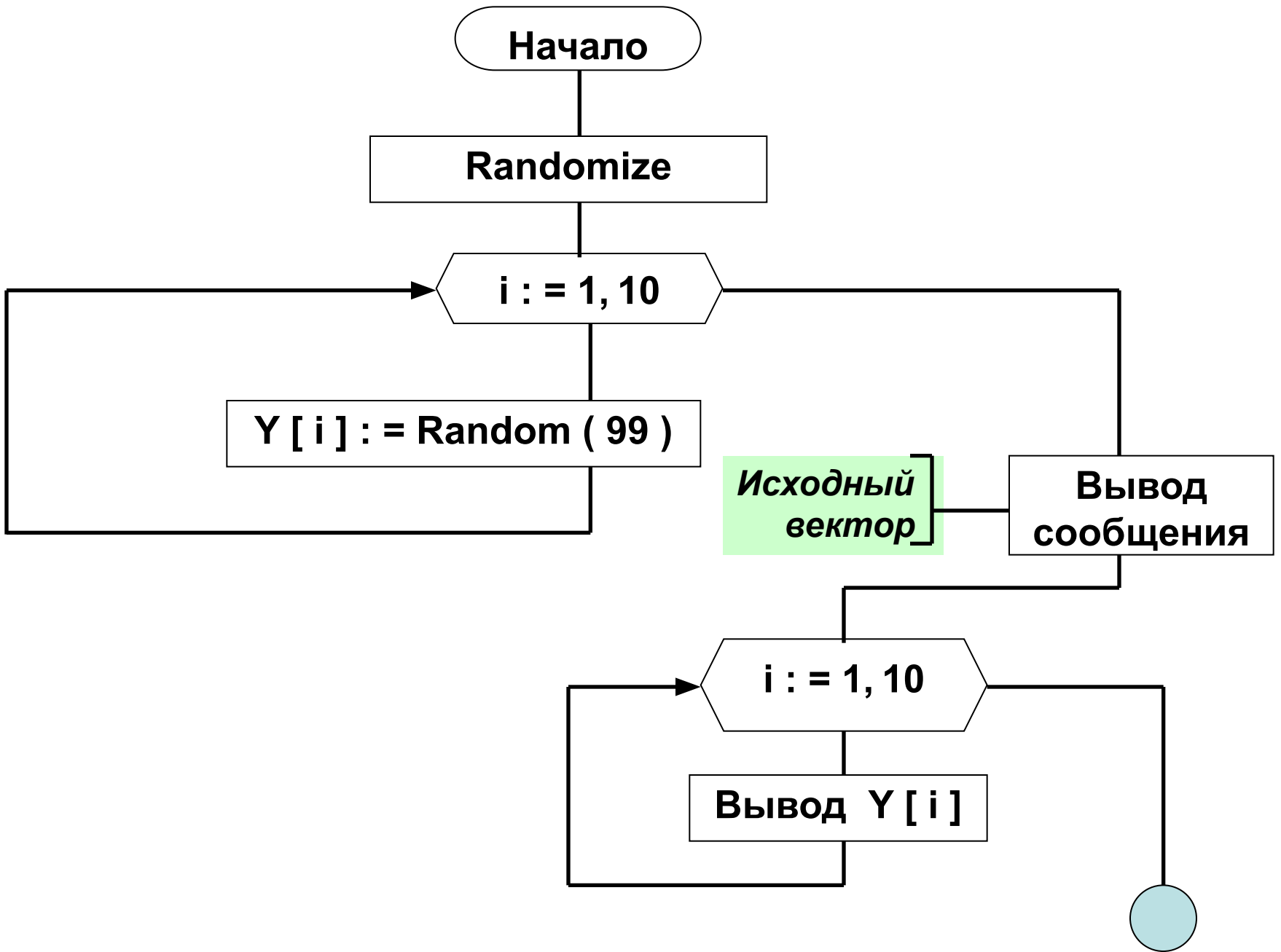
$Y[i] < \text{Min}$

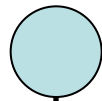
Y =

-2	8	9	1	-8	7	4	-3
-----------	----------	----------	----------	-----------	----------	----------	-----------

i := 1 i := 2 i := 3 i := 4 i := 5 i := 6 i := 7 i := 8

Min =





Min := Y [1]

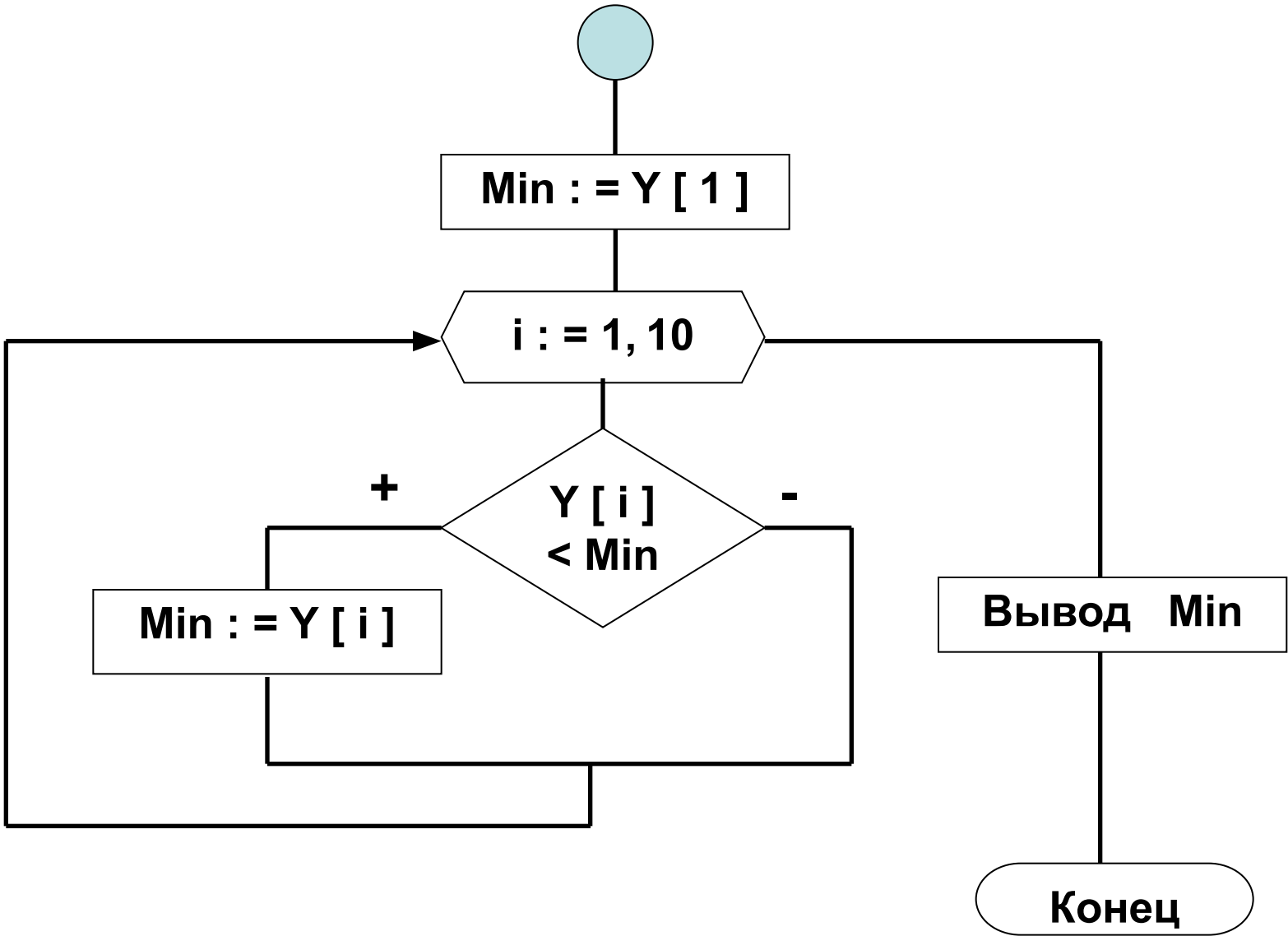
i := 1, 10

+
Y [i] < Min
-

Min := Y [i]

Вывод Min

Конец



Program Vektor ;

Uses CRT ;

Var

Y : array [1 .. 10] of integer ; { вектор }

i : byte ; { индекс элементов вектора }

Min : integer ; { минимальный элемент }

BEGIN

CLrScr ;

Randomize ; { инициализация генератора случайных чисел }

FOR i := 1 **to** 10 **do** { заполнение вектора }

Y [i] := **RANDOM** (99) ;

WriteLn (' Исходный вектор ') ;

FOR i := 1 **to** 10 **do** { вывод вектора на экран }

Write (Y [i] : 5) ; { форматный вывод }

WriteLn ; { переход на новую строку }

Min := Y [1] ;

FOR i := 1 **to** 10 **do** { поиск минимального элемента }
IF Y [i] < Min **THEN** Min := Y [i] ;

WriteLn (' Минимальный элемент вектора = ' , Min) ;

ReadLn; { задержка выполнения программы }

END.

Задача:

Организовать ручное заполнение одномерного массива Z , размерностью 8, вещественными числами.

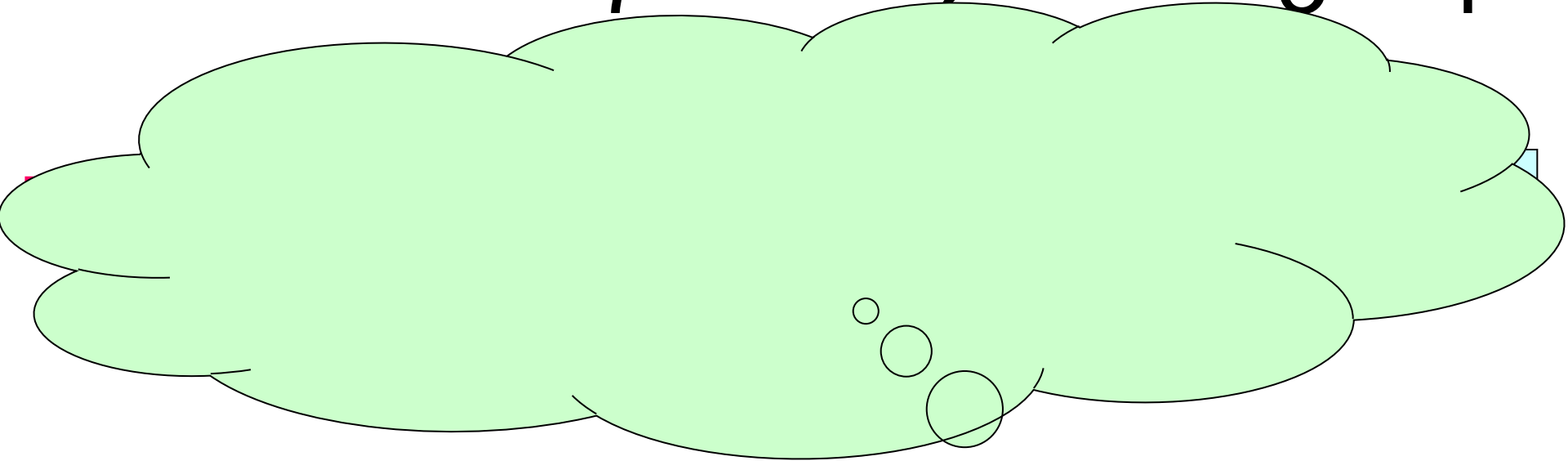
Переставить местами первый элемент массива и элемент массива, равный числу X .

Вывести на экран массив ДО и ПОСЛЕ перестановки.

$$X = 1.1$$

$Z =$

-2.3	8.1	9.	1.1	-8.	7.9	4.	-3.
		7		2		3	1



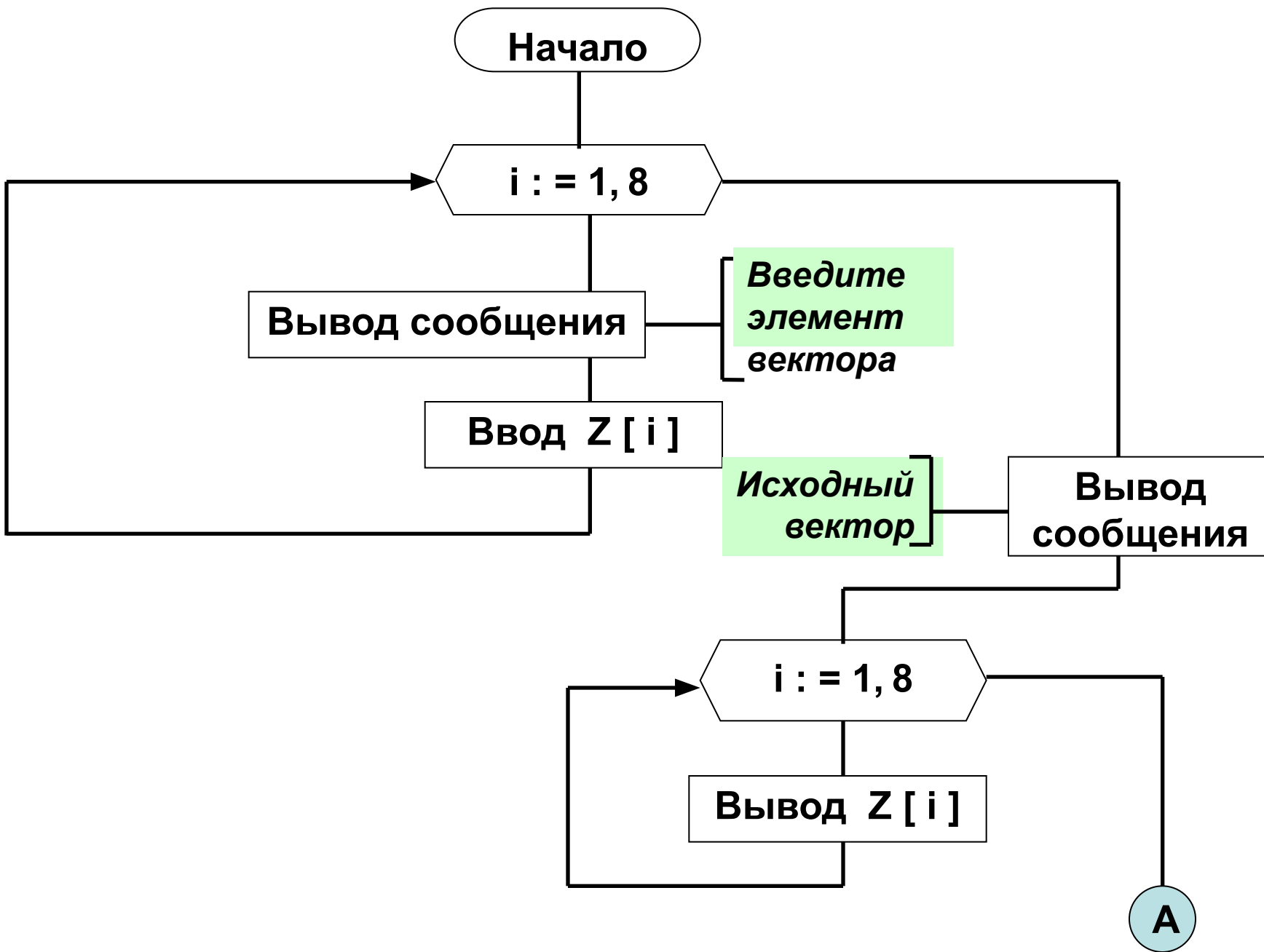
X = 1.1

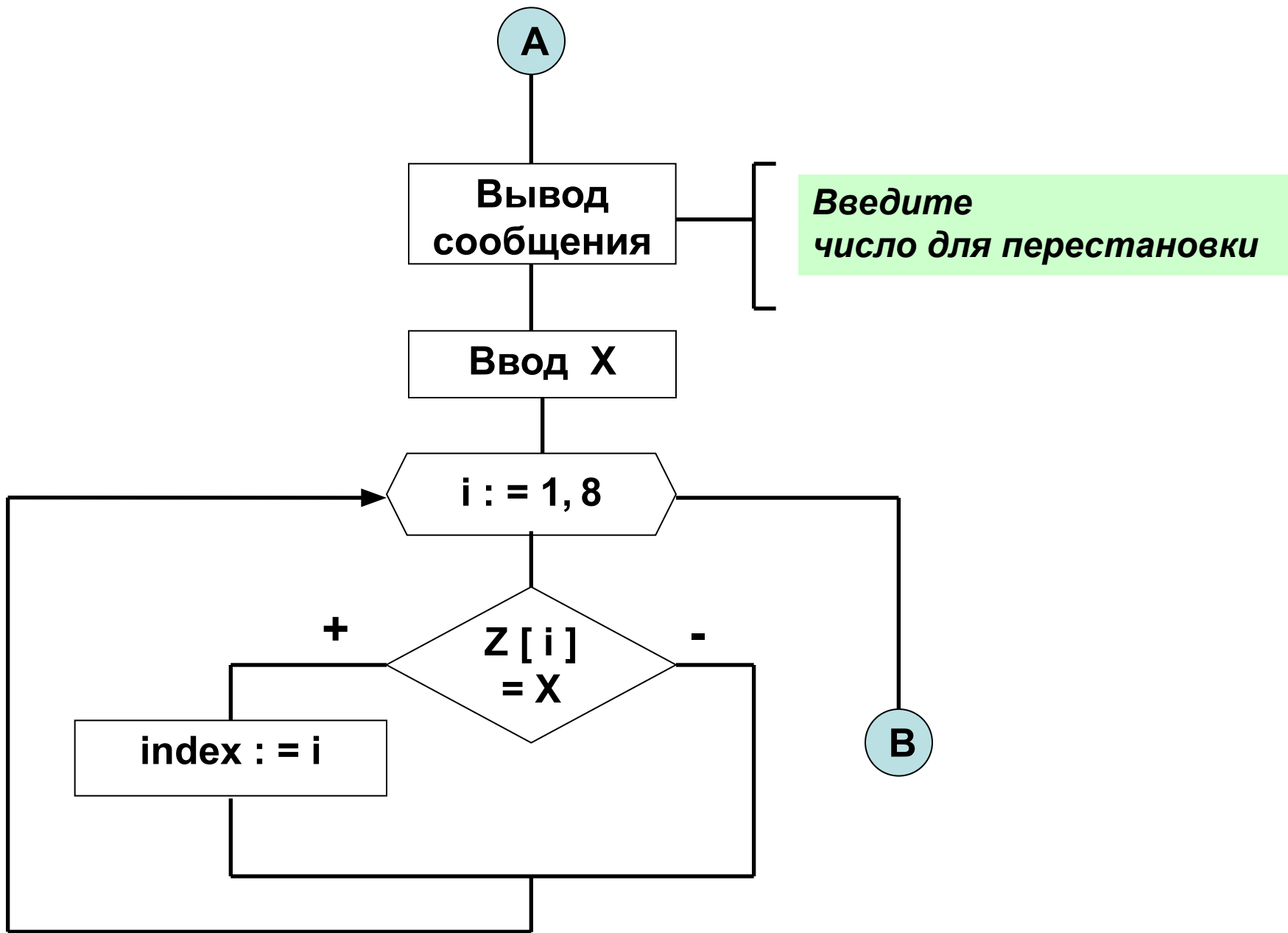
Index = 4

Z =

-2. 3	8.1	9. 7	1. 1	-8.2	7.9	4. 3	-3. 1
------------------------	-----	---------	-----------------------	------	-----	---------	----------

A =





A

**Вывод
сообщения**

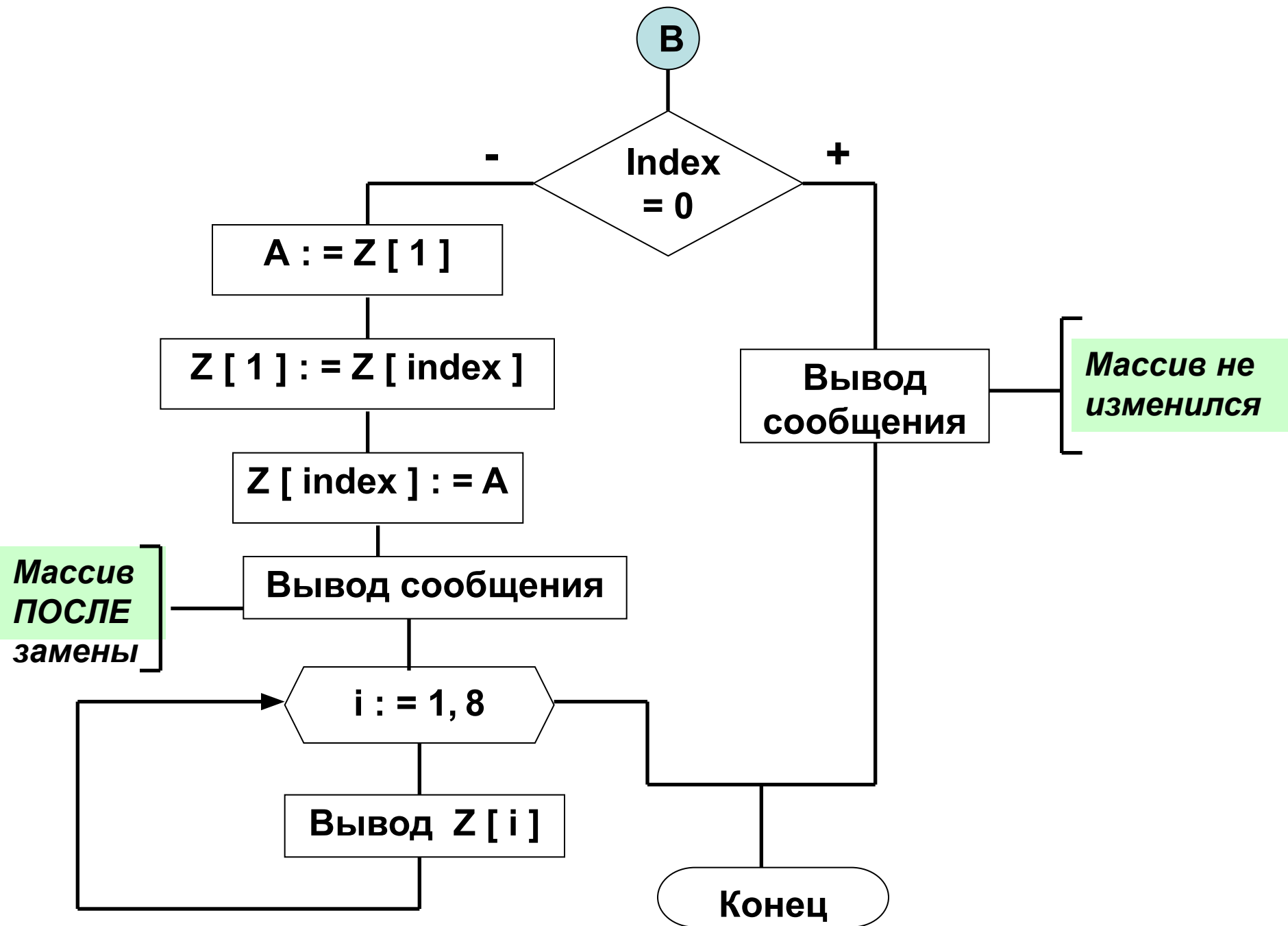
Ввод X

i := 1, 8

**Z[i]
= X**

index := i

B



Program Vektor ;

Var

Z : array [1 .. 8] of REAL ; { вектор }
i : byte ; { индекс элементов вектора }
A : real ; { для перестановки элементов }
X : real ; { число для сравнения }
index : byte ; { индекс переставляемого элемента }

BEGIN

FOR i := 1 to 8 do { заполнение вектора }

begin

Write (' Введите элемент вектора : ') ;

ReadLn (Z [i]) ;

end ;

WriteLn (' Исходный вектор ') ;

FOR i := 1 to 8 do { вывод вектора на экран }

Write (Z [i] : 6 : 1) ; { форматный вывод }

WriteLn ; { переход на новую строку }

```
Write ( ' Введите число для ПЕРЕСТАНОВКИ : ' ) ;  
ReadLn ( X ) ;
```

```
FOR i := 1 to 8 do { поиск индекса переставляемого элемента }  
IF Z [ i ] = X THEN index := i ;
```

```
IF index = 0 THEN WriteLn ( ' Вектор НЕ изменился ' )  
ELSE
```

```
begin
```

```
  A := Z [ 1 ] ;
```

```
  Z [ 1 ] := Z [ index ] ;
```

```
  Z [ index ] := A ;
```

```
  WriteLn ( ' Вектор ПОСЛЕ перестановки элементов ' ) ;
```

```
  FOR i := 1 to 8 do { вывод вектора на экран }
```

```
    Write ( Z [ i ] : 6 : 1 ) ; { форматный вывод }
```

```
end ;
```

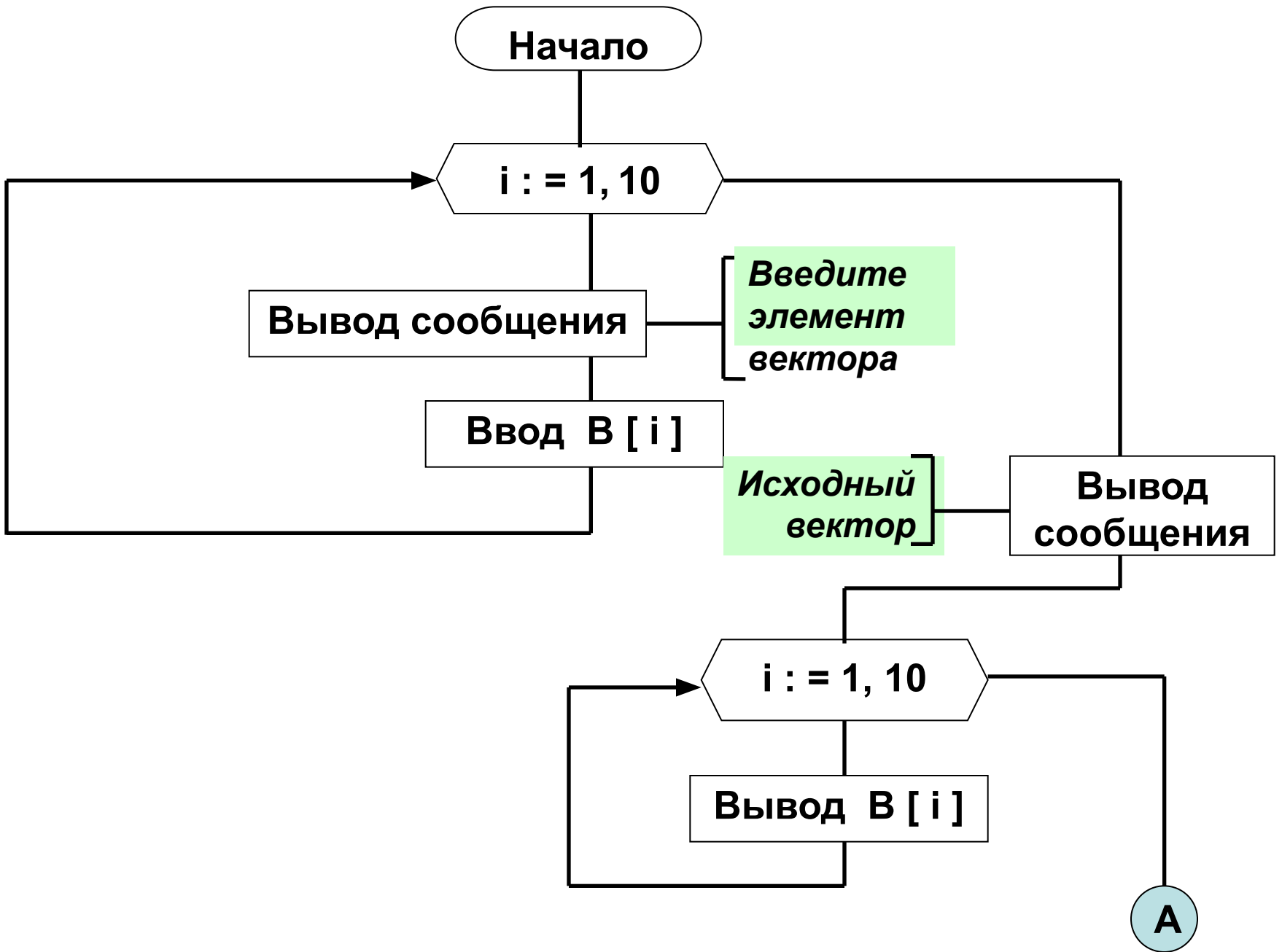
```
ReadLn; { задержка выполнения программы }
```

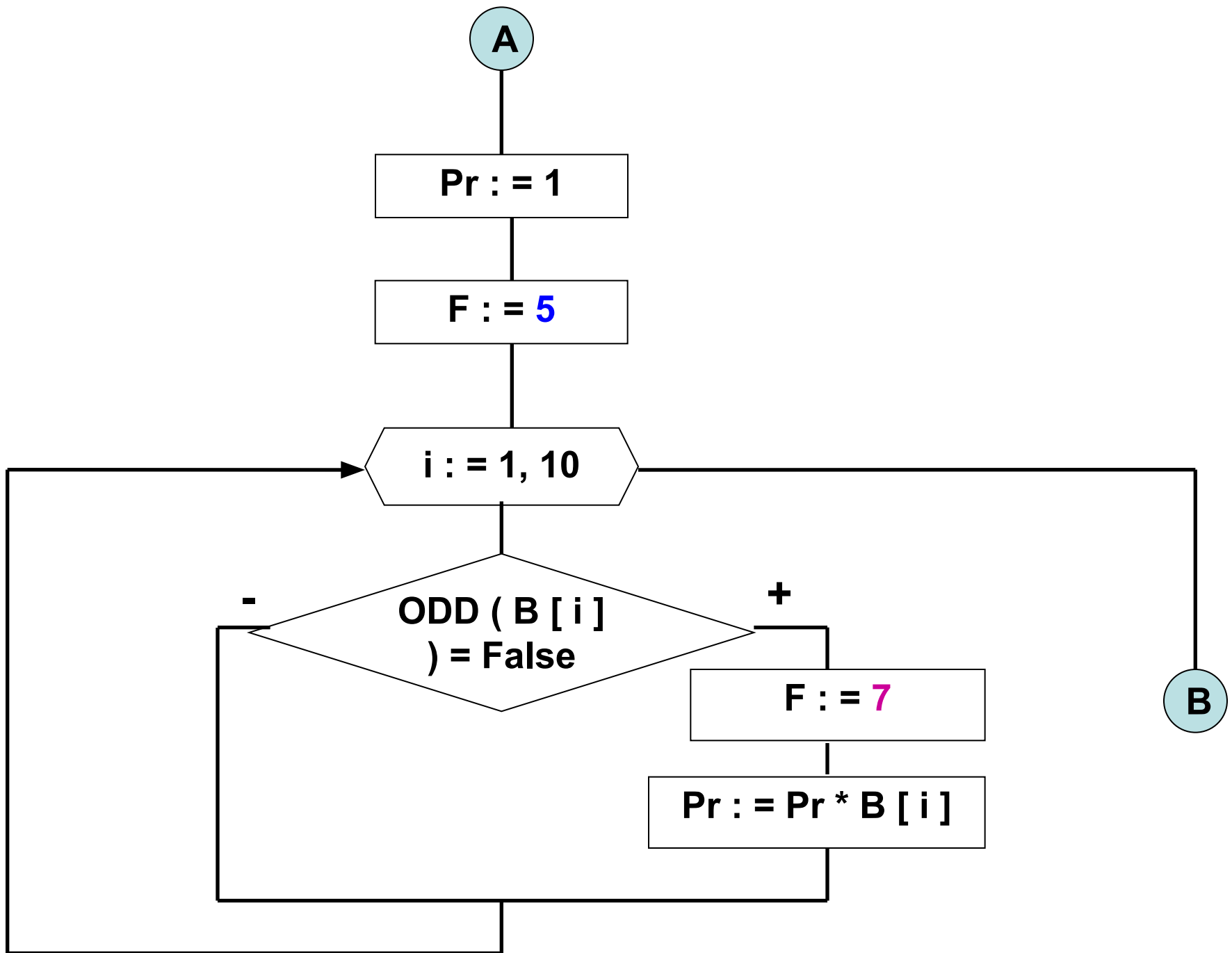
```
END.
```

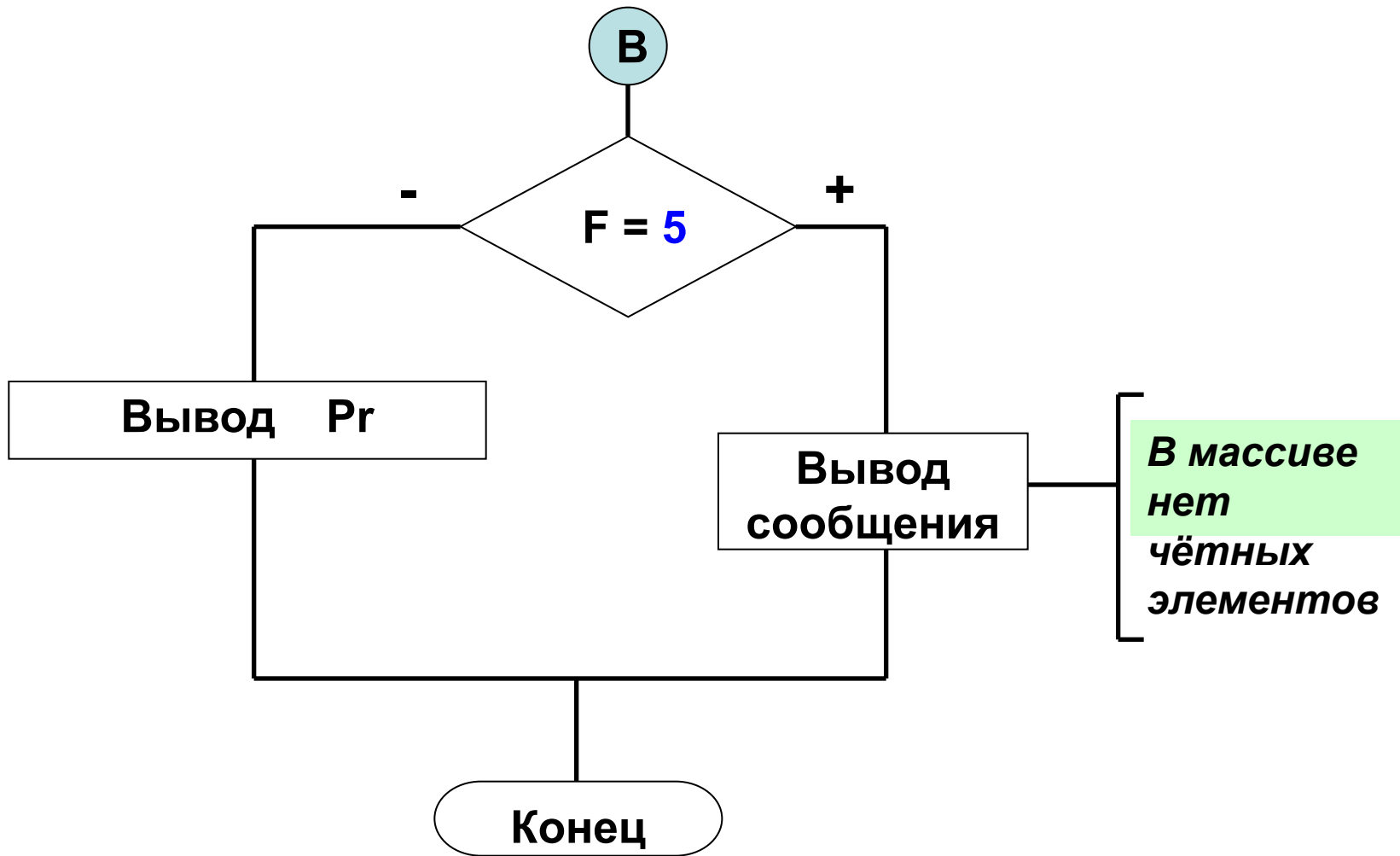
Задача:

Организовать ручное заполнение одномерного массива В, размерностью 10, целыми числами.

Найти и вывести на экран **произведение** чётных элементов массива.







Program Vektor ;

Var

B : array [1 .. 10] of integer ; { вектор }

i : byte ; { индекс элементов вектора }

F : byte ; { переключатель }

Pr : integer ; { произведение чётных элементов }

BEGIN

FOR i := 1 **to** 10 **do** { заполнение вектора }

begin

Write (' Введите элемент вектора : ') ;

ReadLn (B [i]) ;

end ;

WriteLn (' Исходный вектор ') ;

FOR i := 1 **to** 10 **do** { вывод вектора на экран }

Write (B [i] : 6) ; { форматный вывод }

WriteLn ; { переход на новую строку }

F := 5 ;

Pr := 1 ; { первоначальное значение произведения }

FOR i := 1 to 10 do { поиск произведения чётных элементов }

IF ODD (B [i]) = FALSE THEN begin

F := 7 ;

Pr := Pr * B [i] ;

end ;

IF F = 5

THEN WriteLn (' В массиве НЕТ чётных элементов ')

ELSE

WriteLn (' Произведение чётных элементов равно ' , Pr) ;

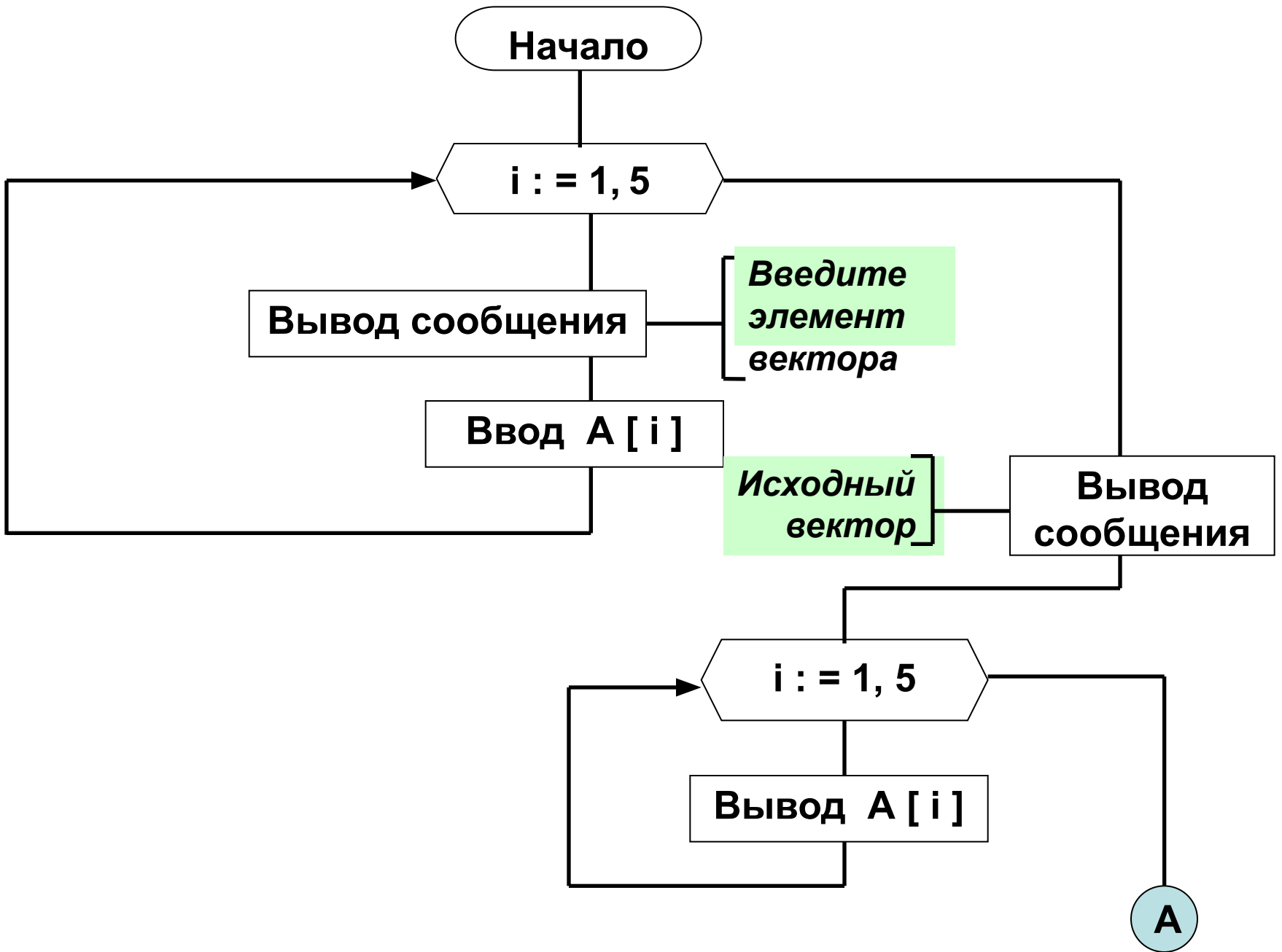
ReadLn; { задержка выполнения программы }

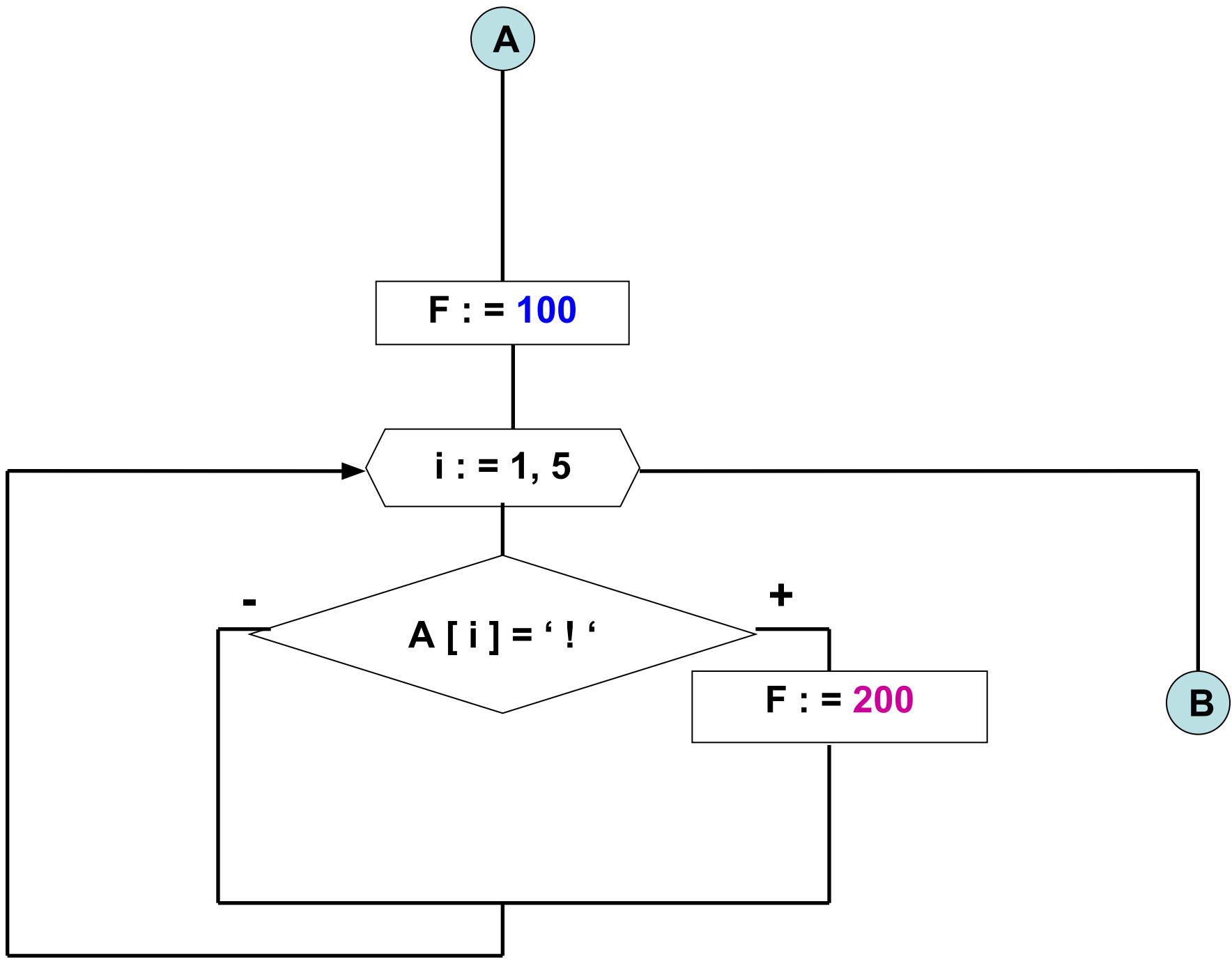
END.

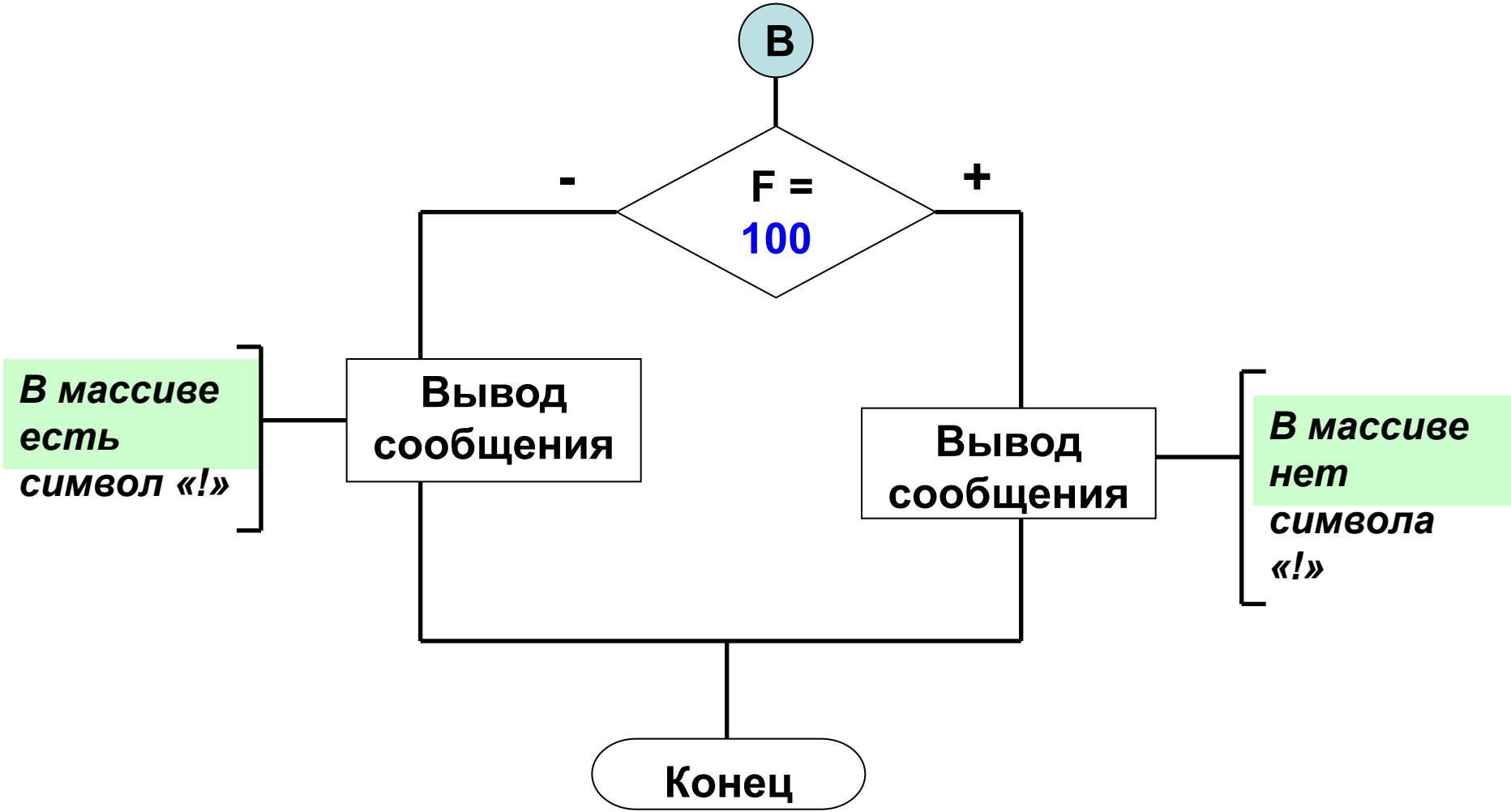
Задача:

Организовать **ручное** заполнение символьного одномерного массива А, размерностью 5.

Определить и вывести на экран **сообщение** о наличии в массиве символа «!».







Program Vektor ;

Var

A : array [1 .. 5] of char ; { *символьный вектор* }

i : byte ; { *индекс элементов вектора* }

F : byte ; { *переключатель* }

BEGIN

FOR i := 1 to 5 do { *заполнение вектора* }

begin

Write (' Введите элемент вектора : ') ;

ReadLn (A [i]) ;

end ;

WriteLn (' Исходный вектор ') ;

FOR i := 1 to 5 do { *вывод вектора на экран* }

Write (A [i] : 6) ; { *форматный вывод* }

WriteLn ; { *переход на новую строку* }

F := 100 ;

FOR i := 1 **to** 5 **do** { поиск символа «!» }
IF A[i] = '!' **THEN** F := 200 ;

IF F = 100

THEN WriteLn (' В массиве НЕТ символа «!» ')

ELSE WriteLn (' В массиве ЕСТЬ символ «!» ') ;

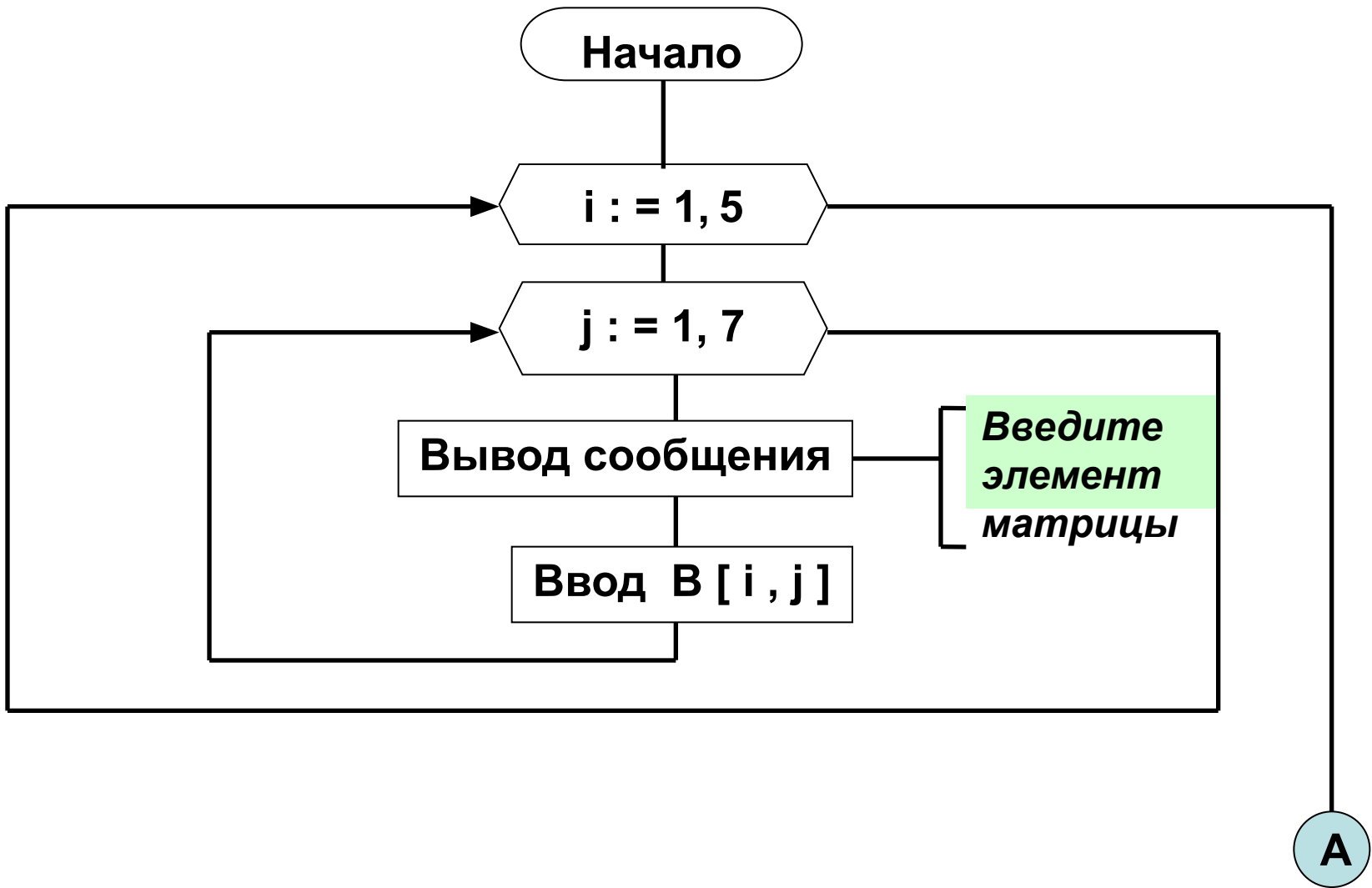
ReadLn; { задержка выполнения программы }

END.

Задача:

Организовать **ручное** заполнение двумерного массива В, размерностью 5 x 7, целыми числами.

Найти и вывести на экран **количество** чётных элементов массива и **максимальный** элемент матрицы.



A

Вывод сообщения

Исходная матрица

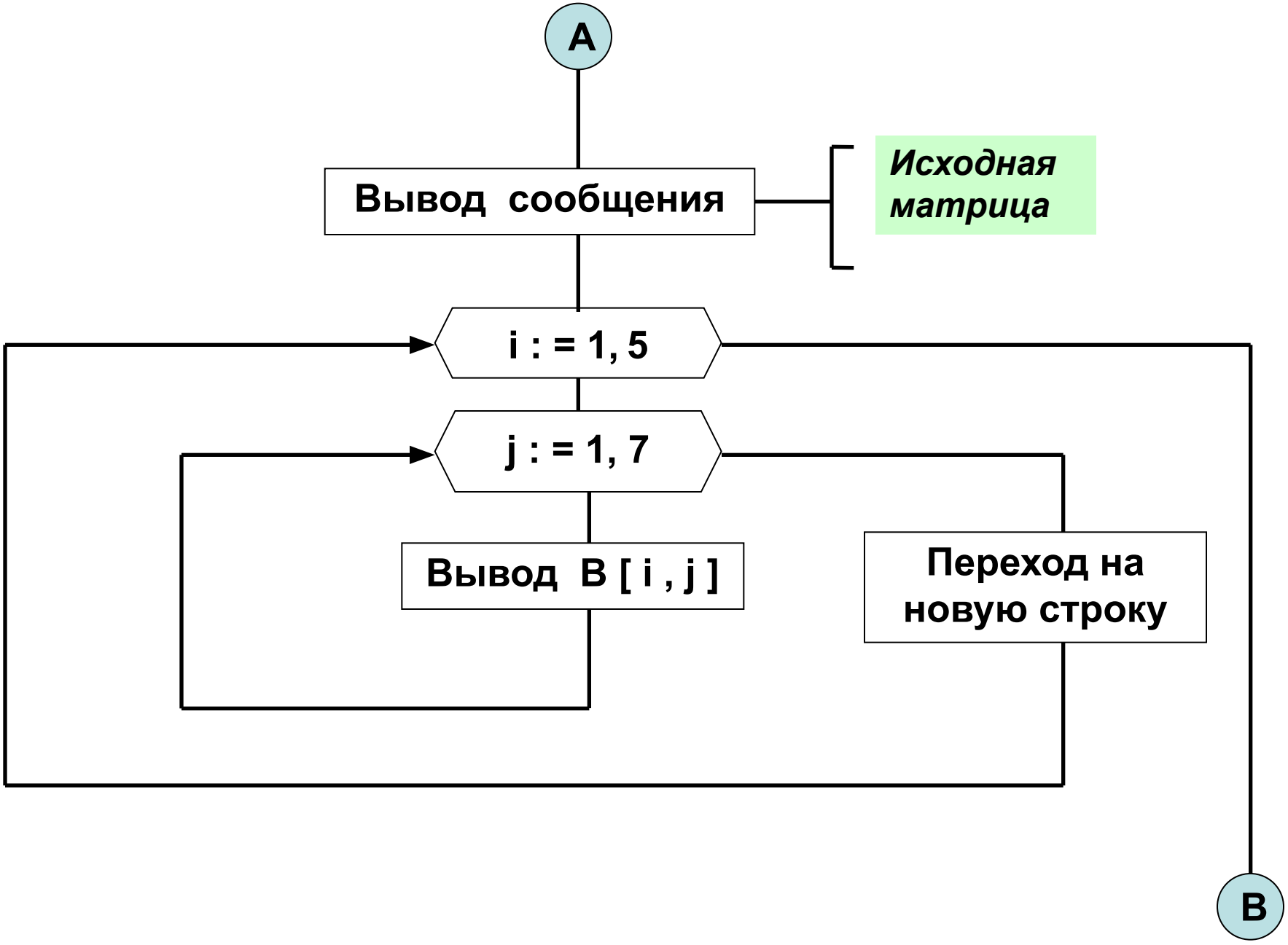
$i := 1, 5$

$j := 1, 7$

Вывод $V[i, j]$

Переход на новую строку

B

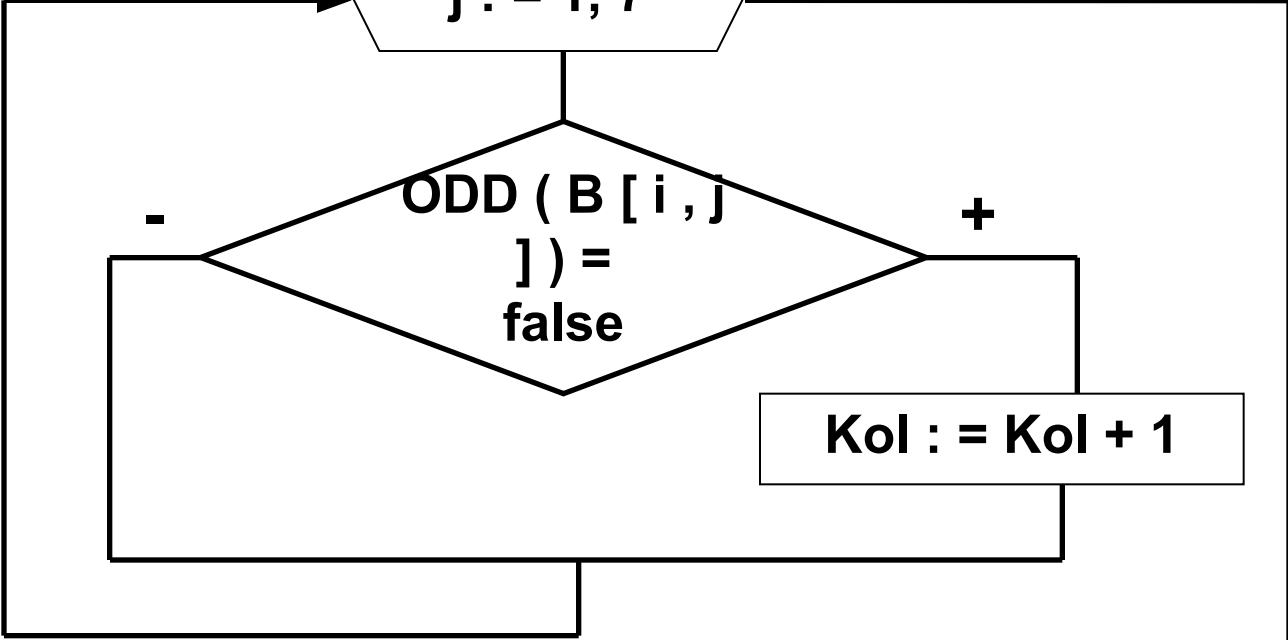


B

Kol := 0

i := 1, 5

j := 1, 7



**Вывод
Kol**

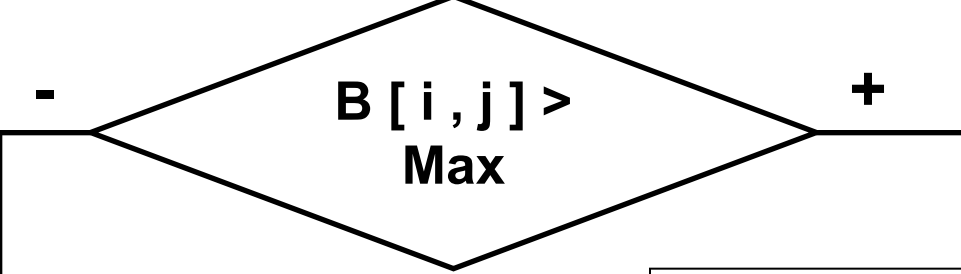
C

С

Max := B [1, 1]

i := 1, 5

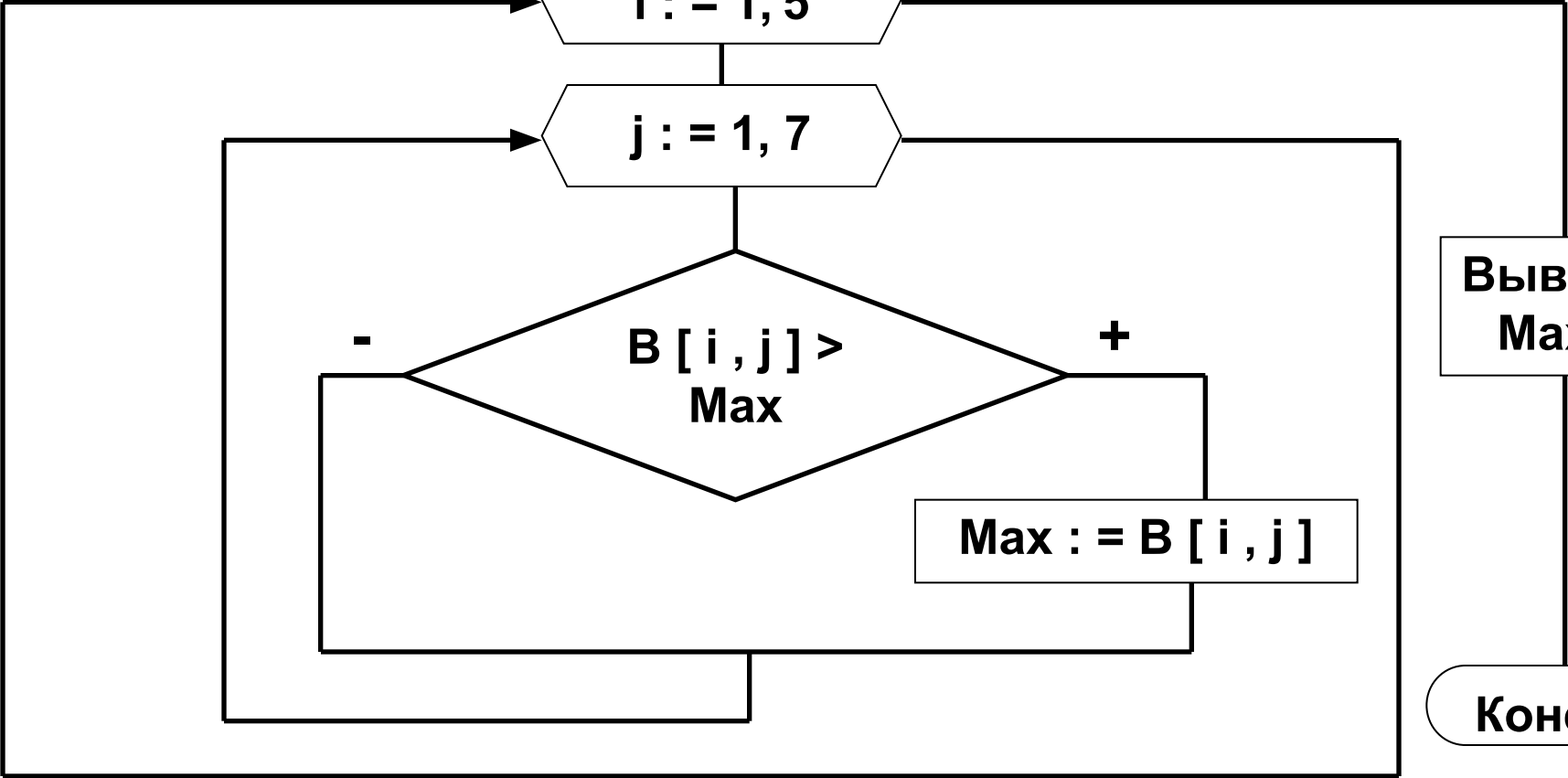
j := 1, 7



Max := B [i, j]

Вывод
Max

Конец



Program Matrica ;

Var

B : array [1 .. 5 , 1 .. 7] of integer ; { матрица }

i , j : byte ; { индексы элементов матрицы }

KoL : byte; { кол-во чётных элементов }

Max : integer ; { максимальный элемент }

BEGIN

FOR i := 1 to 5 do { заполнение матрицы }

FOR j := 1 to 7 do

begin

Write (' Введите элемент матрицы ->... ');

ReadLn (B [i , j]);

end ;

```
WriteLn ( ' Исходная матрица ' ) ;
```

```
FOR i := 1 to 5 do { вывод матрицы на экран }
```

```
begin
```

```
FOR j := 1 to 7 do
```

```
Write ( B [ i , j ] : 6 ) ; { форматный вывод }
```

```
WriteLn ; { переход на новую строку }
```

```
end ;
```

```
KoL := 0 ;
```

```
FOR i := 1 to 5 do { поиск кол-ва чётных эл-ов }
```

```
FOR j := 1 to 7 do
```

```
IF ODD ( B [ i , j ] ) = faLse THEN KoL := KoL + 1 ;
```

```
WriteLn ( ' Кол-во чётных элементов равно ' , KoL ) ;
```

```
Max := B [ 1 , 1 ] ;
```

```
FOR i := 1 to 5 do { поиск max элемента }
```

```
FOR j := 1 to 7 do
```

```
IF B [ i , j ] > Max THEN Max := B [ i , j ] ;
```

```
WriteLn ( ' Максимальный элемент равен ' , Max ) ;
```

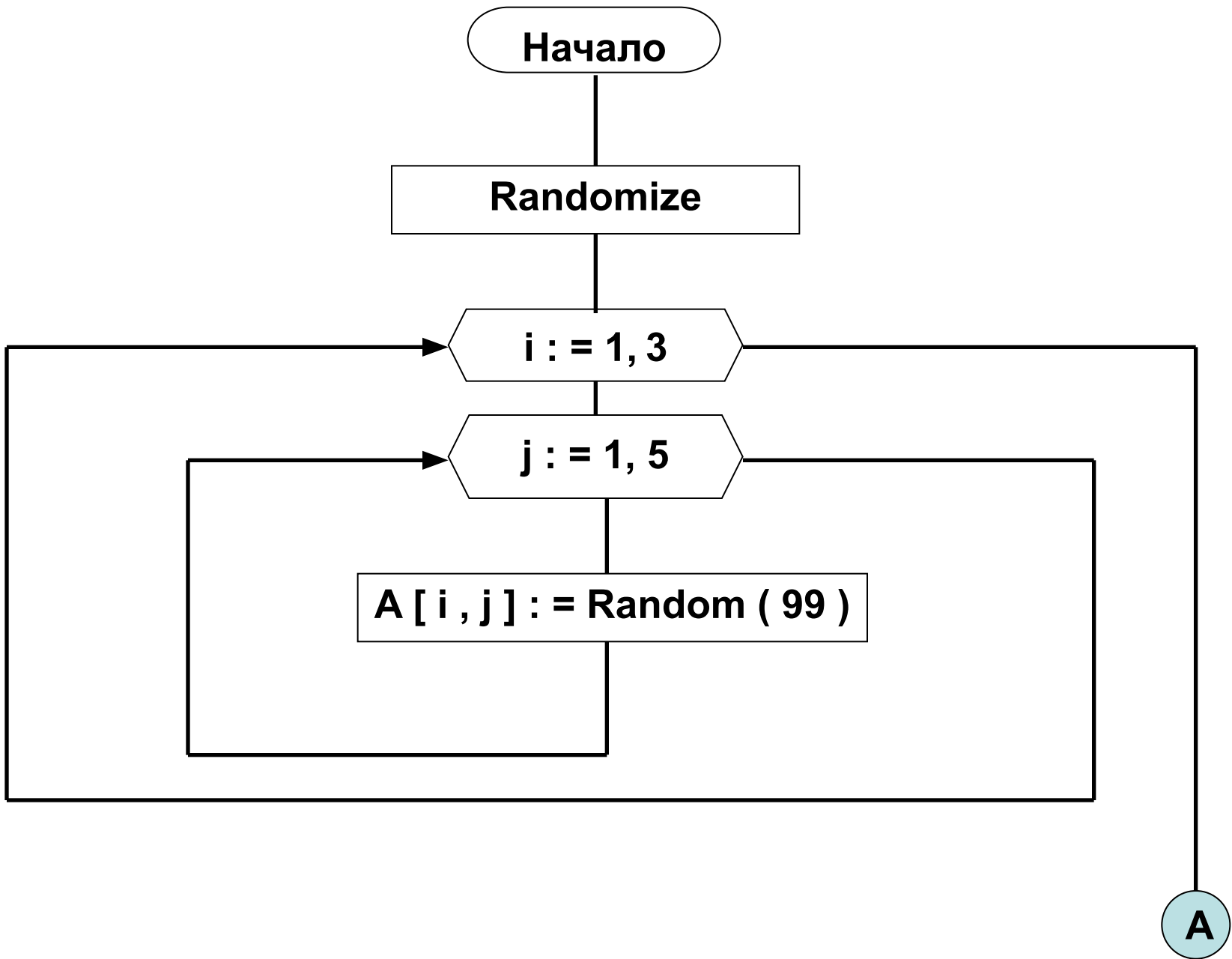
```
ReadLn ;
```

```
END.
```

Задача:

Организовать **случайное** заполнение двумерного массива A , размерностью 3×5 , целыми числами.

Найти и вывести на экран **произведение** нечётных элементов массива в столбце T .



A

Вывод сообщения

Исходная матрица

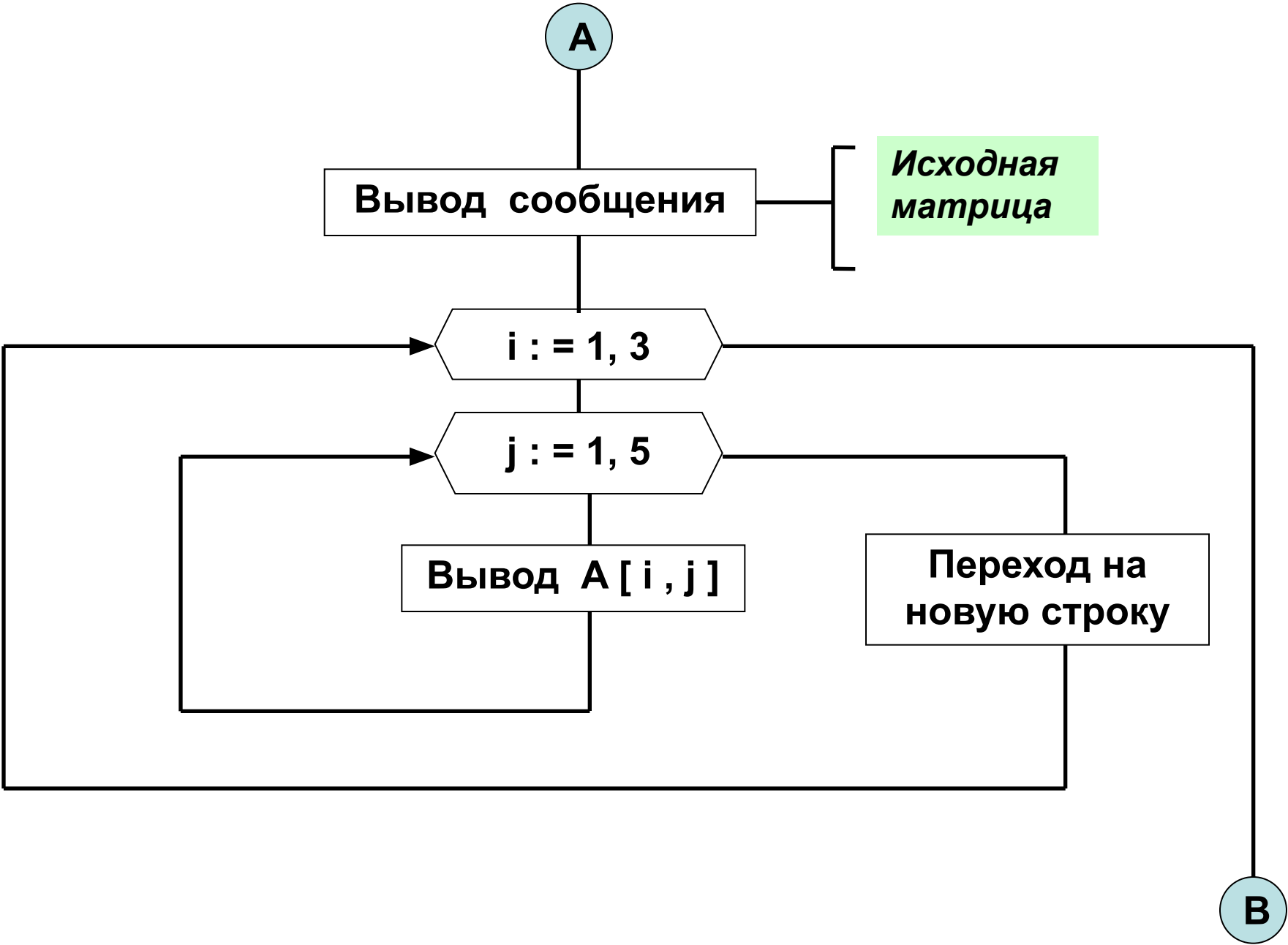
$i := 1, 3$

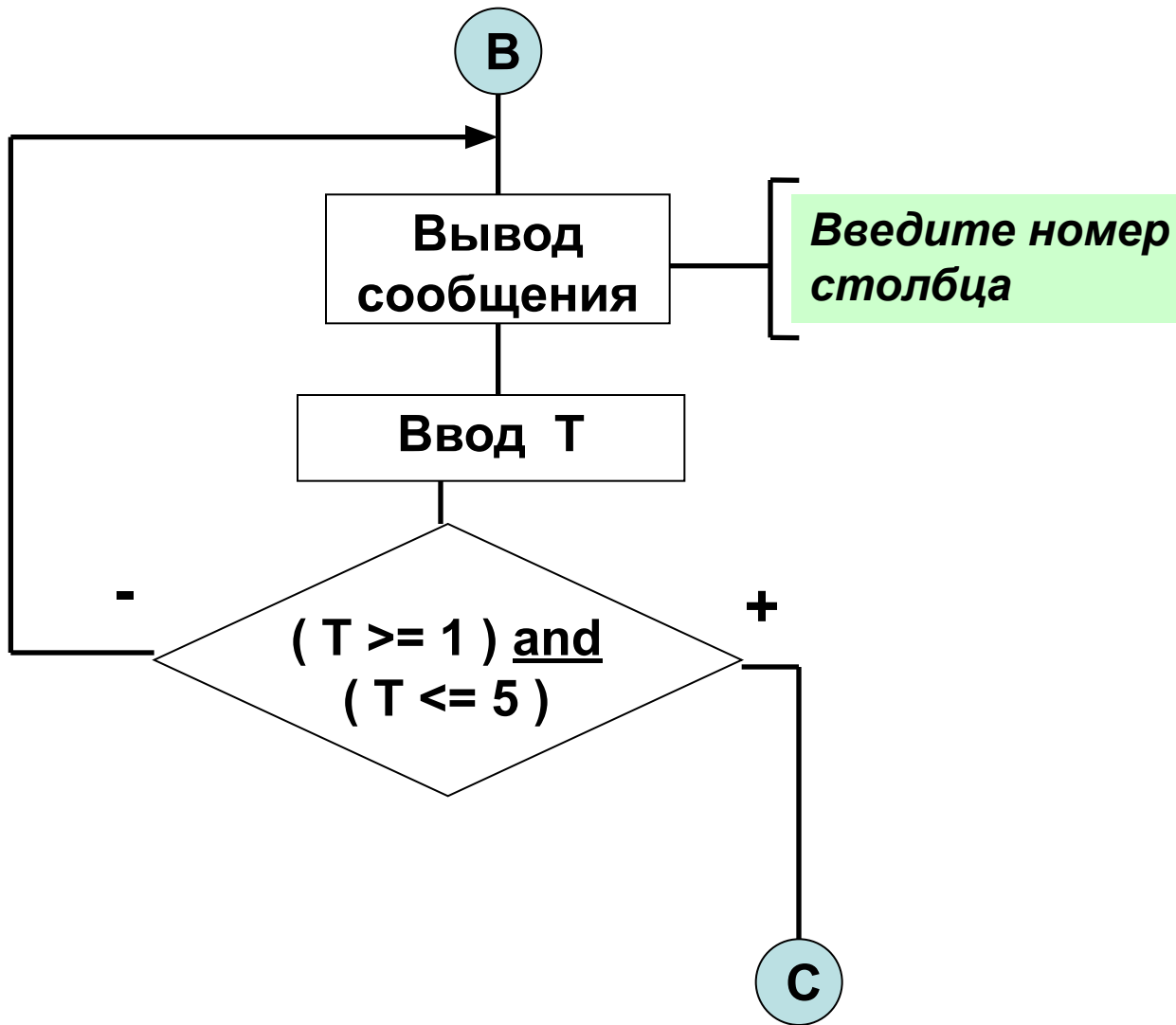
$j := 1, 5$

Вывод $A[i, j]$

Переход на новую строку

B





C

P := 1

F := 2

i := 1, 3

ODD (A [i ,
T]) =
true

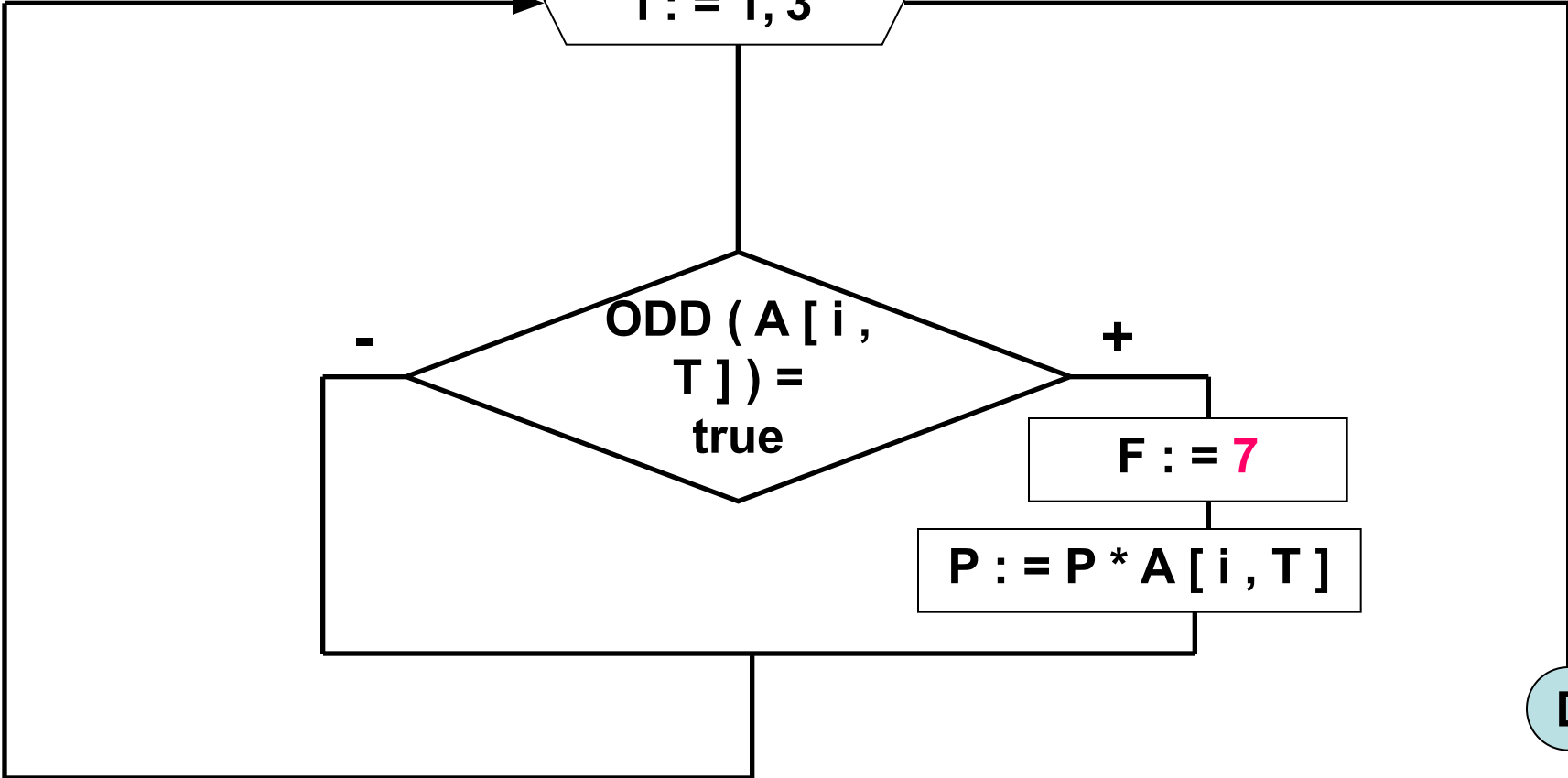
-

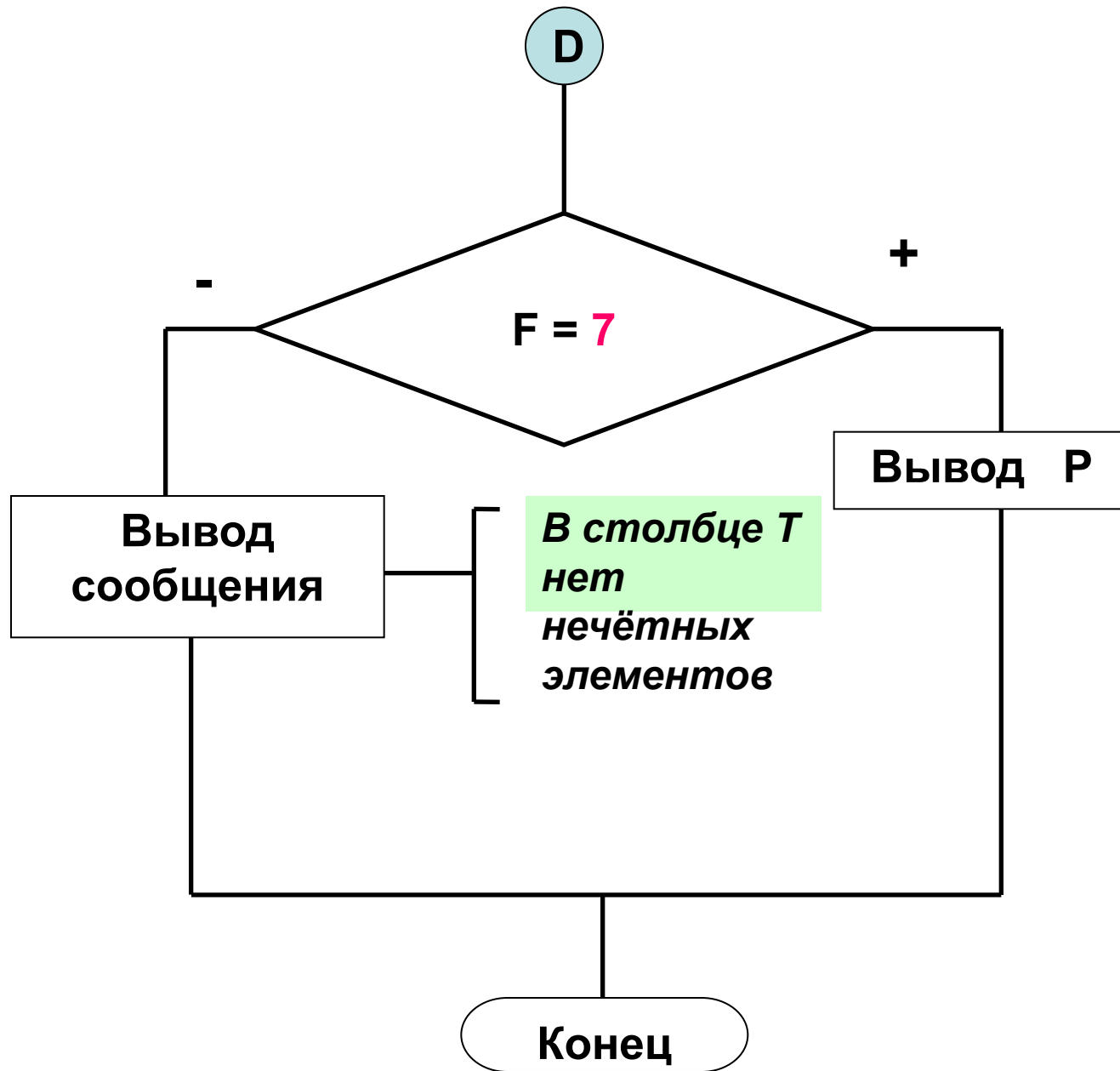
+

F := 7

P := P * A [i , T]

D





Program Matrica ;

Var

A : array [1 .. 3 , 1 .. 5] of integer ; { матрица }

i , j : byte ; { индексы элементов матрицы }

P : integer ; { произведение нечётных эл-ов }

T : integer ; { номер столбца }

F : integer ; { переключатель }

BEGIN

Randomize ; { инициализация генератора случайных чисел }

FOR i := 1 to 3 do { заполнение матрицы }

FOR j := 1 to 5 do

A [i , j] := Random (99) ;

```
WriteLn ( ' Исходная матрица ' ) ;
```

```
FOR i := 1 to 3 do { вывод матрицы на экран }
```

```
begin
```

```
FOR j := 1 to 5 do
```

```
Write ( A [ i , j ] : 6 ) ; { форматный вывод }
```

```
WriteLn ; { переход на новую строку }
```

```
end ;
```

```
REPEAT { проверка корректности ввода столбца }
```

```
WriteLn ( ' Введите номер столбца ' ) ;
```

```
ReadLn ( T ) ;
```

```
UNTIL ( T > = 1 ) and ( T < = 5 ) ;
```

```
P := 1 ;
```

```
F := 2 ;
```

FOR $i := 1$ **to** 3 **do** { поиск произведения }

IF $\text{ODD}(A[i, T]) = \text{true}$ **THEN** **begin**

$F := 7$;

$P := P * A[i, T]$;

end ;

IF $F = 7$ **THEN**

WriteLn (' Произведение нечётных элементов = ' , P)

ELSE

WriteLn (' В столбце T нет нечётных элементов ') ;

ReadLn ;

END .

$$A_{3 \times 3} = \begin{bmatrix} 2 & 5 & 7 \\ 8 & -3 & 4 \\ 1 & 9 & -2 \end{bmatrix}$$

5 \longrightarrow **A[1, 2]**

7 \longrightarrow **A[1, 3]**

4 \longrightarrow **A[2, 3]**

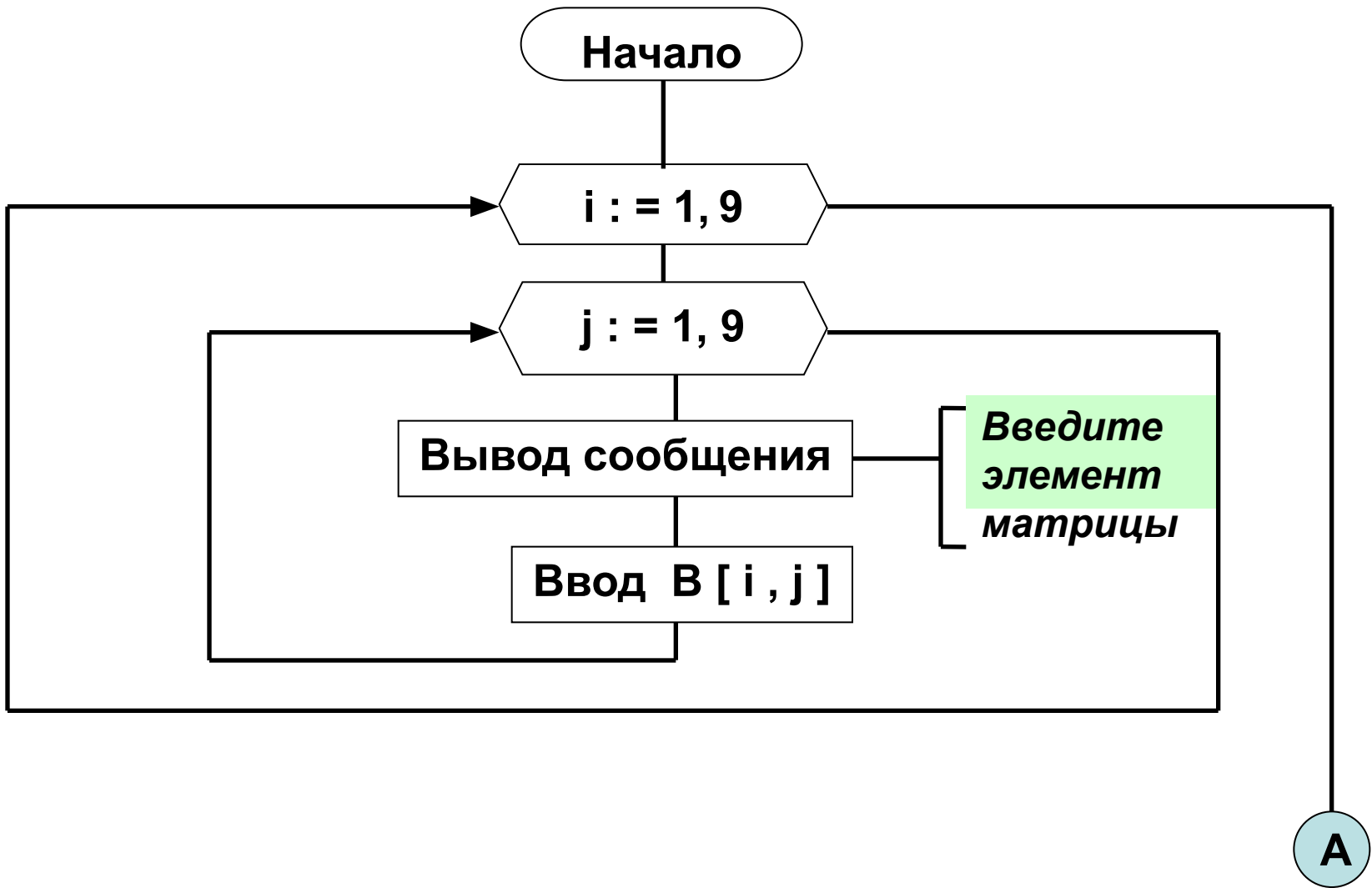
i < **j**

Задача:

Организовать **ручное** заполнение двумерного массива В, размерностью 9 x 9, целыми числами.

Увеличить все элементы главной диагонали в два раза.

Вывести на экран массив В ДО и ПОСЛЕ изменения.



A

Вывод сообщения

*Матрица
ДО изменения*

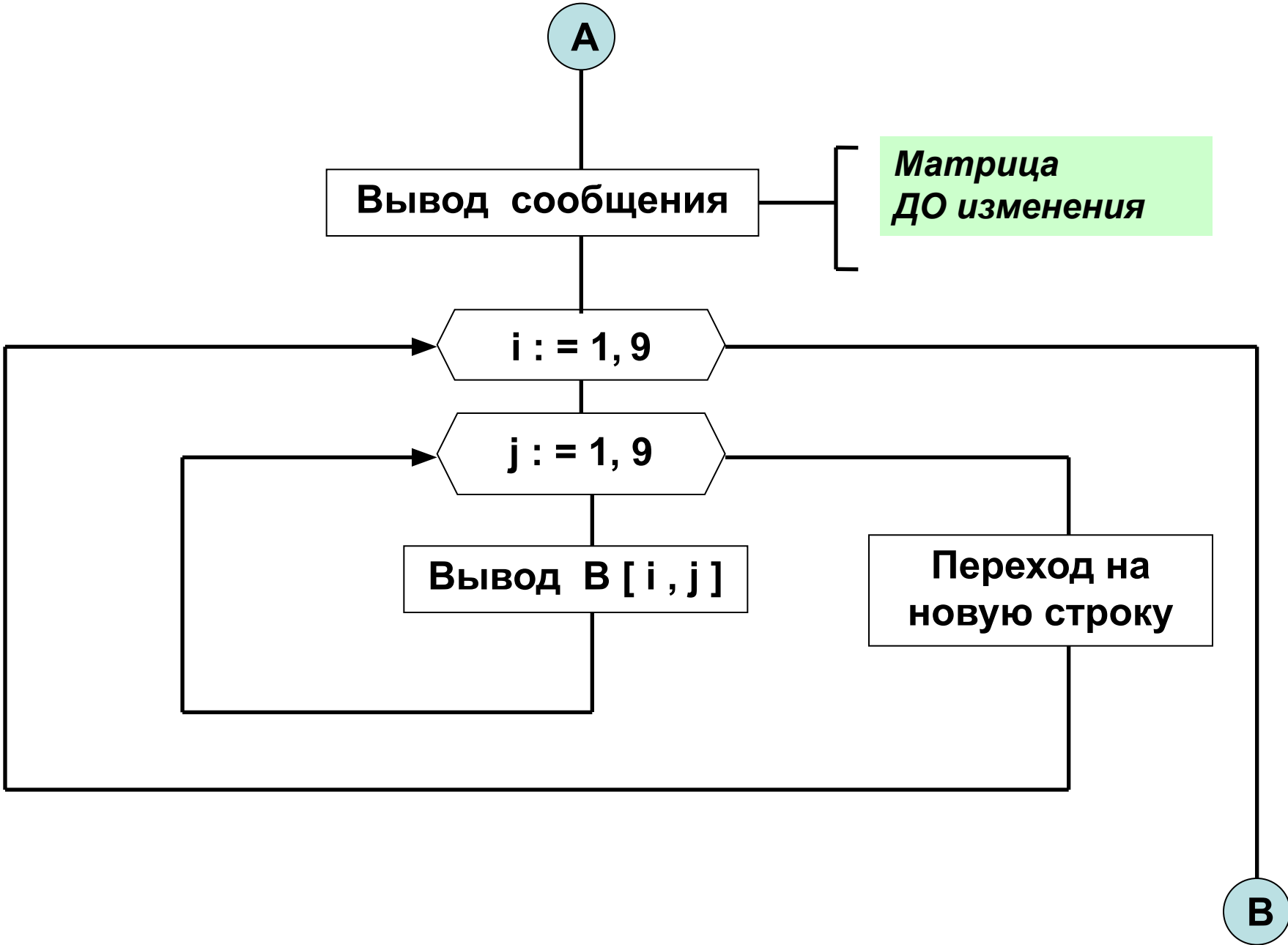
$i := 1, 9$

$j := 1, 9$

Вывод $V[i, j]$

**Переход на
новую строку**

B

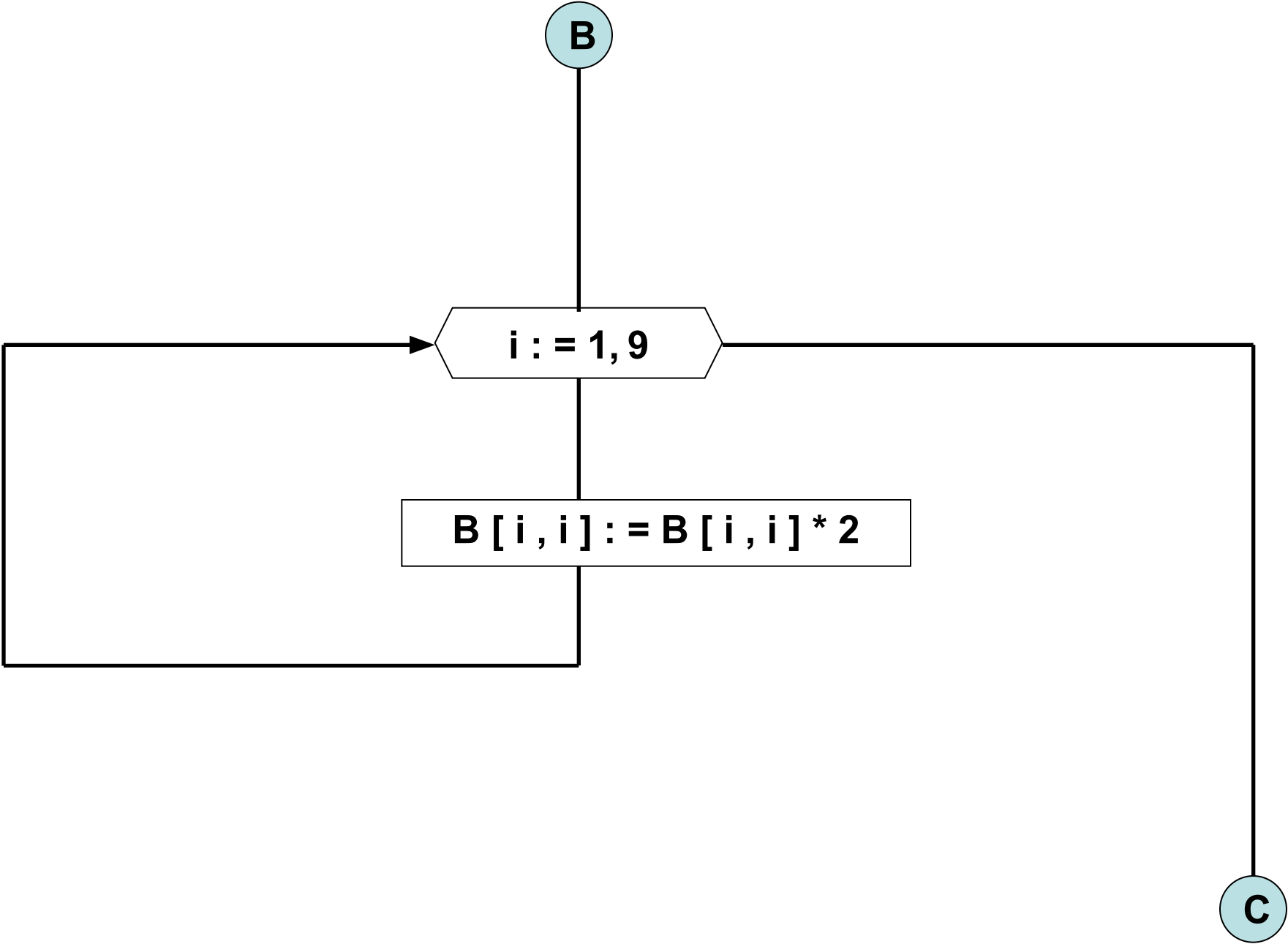


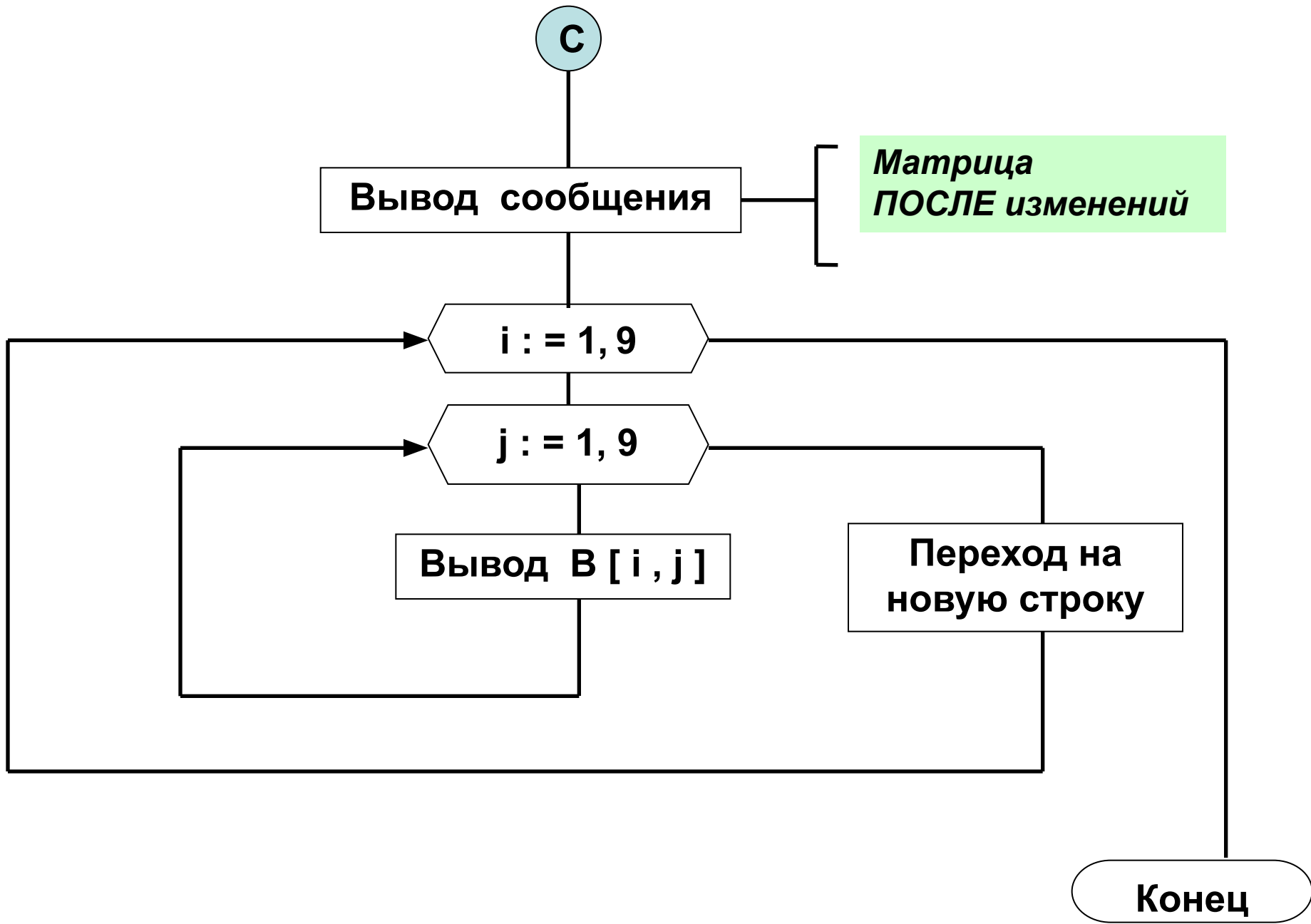
B

$i := 1, 9$

$B[i, i] := B[i, i] * 2$

C





Program Matrica ;

Var

B : array [1 .. 9 , 1 .. 9] of integer ; { матрица }
i , j : byte ; { индексы элементов матрицы }

BEGIN

FOR i := 1 to 9 do { заполнение матрицы }

FOR j := 1 to 9 do

begin

Write (' Введите элемент матрицы : ') ;

ReadLn (B [i , j]) ;

end ;

```
WriteLn ( ' Матрица ДО изменения ' ) ;
```

```
FOR i := 1 to 9 do { вывод матрицы на экран }
```

```
begin
```

```
  FOR j := 1 to 9 do
```

```
    Write ( B [ i , j ] : 6 ) ;
```

```
    WriteLn ;
```

```
end ;
```

```
FOR i := 1 to 9 do { замена элементов диагонали }
```

```
B [ i , i ] := B [ i , i ] * 2 ;
```



```
WriteLn ( ' Матрица ПОСЛЕ изменения ' ) ;
```

```
FOR i := 1 to 9 do { вывод матрицы на экран }
```

```
begin
```

```
  FOR j := 1 to 9 do
```

```
    Write ( B [ i , j ] : 6 ) ;
```

```
    WriteLn ;
```

```
end ;
```

```
ReadLn ;
```

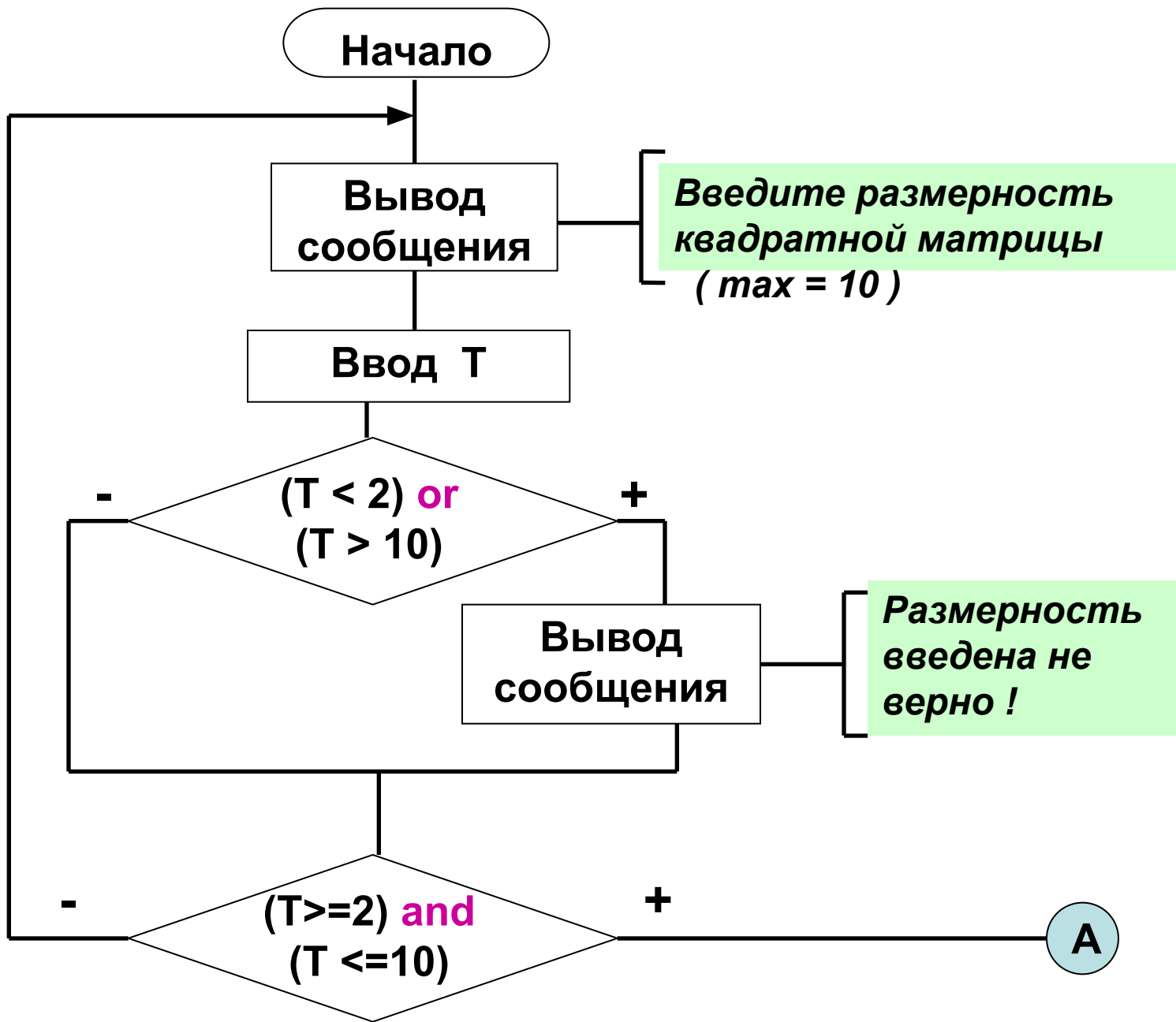
```
END .
```

Задача:

Организовать **случайное** заполнение двумерного массива M , размерностью $T \times T$, целыми числами.

Заменить все элементы **выше** главной диагонали на первый элемент матрицы.

Вывести на экран массив M ДО и ПОСЛЕ изменения.



A

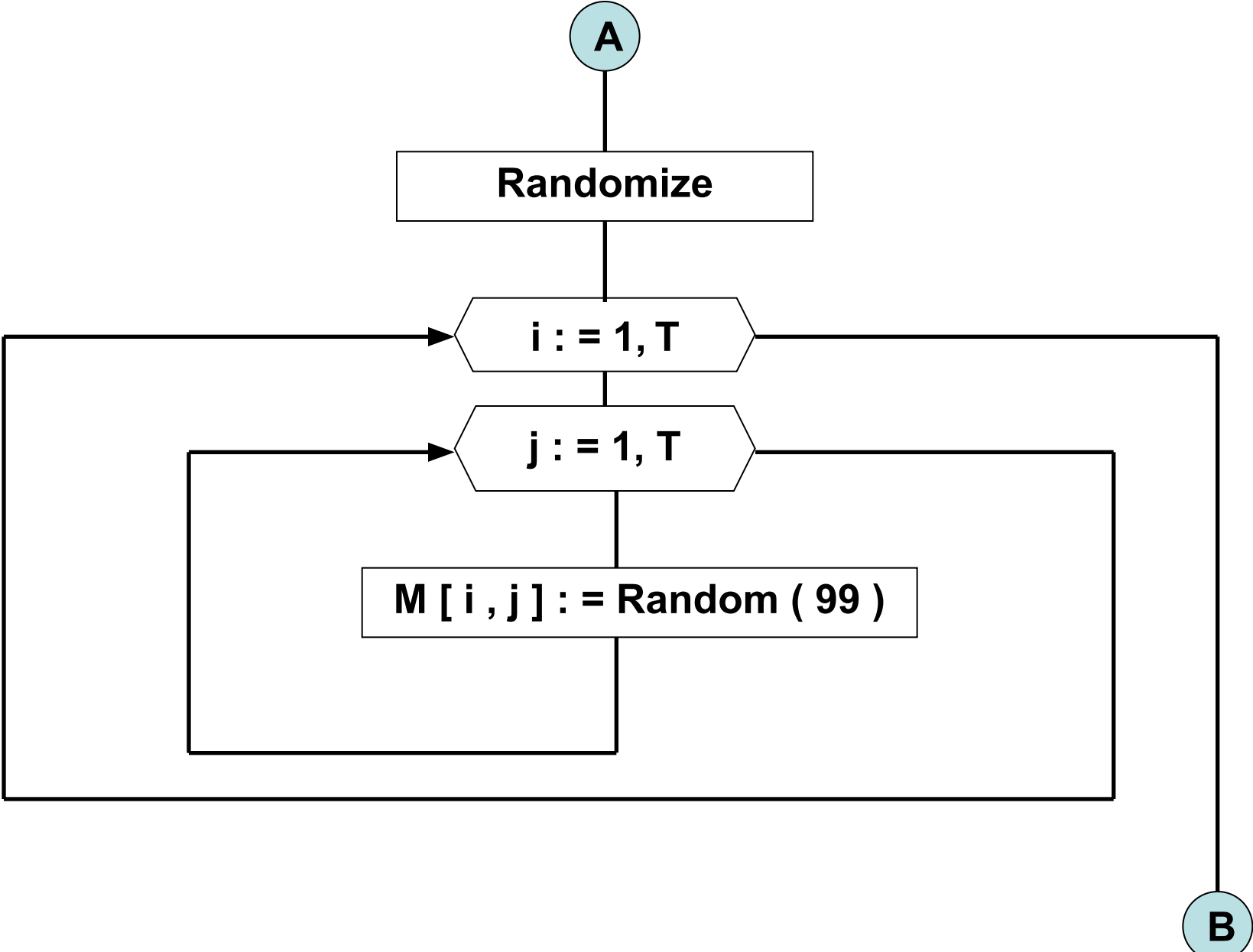
Randomize

$i := 1, T$

$j := 1, T$

$M[i, j] := \text{Random}(99)$

B



В

Вывод сообщения

*Матрица
ДО изменения*

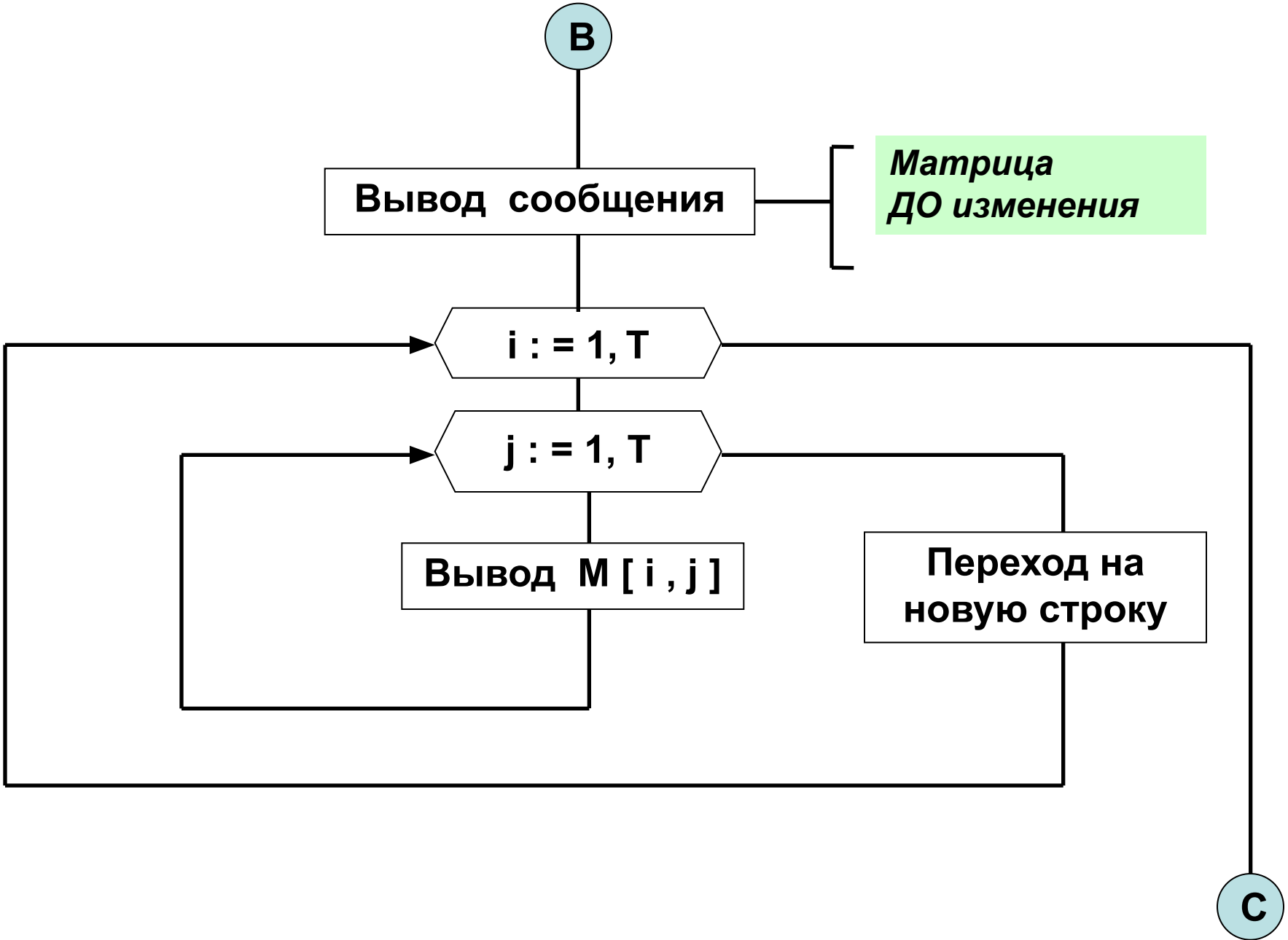
$i := 1, T$

$j := 1, T$

Вывод $M[i, j]$

Переход на
новую строку

С



C

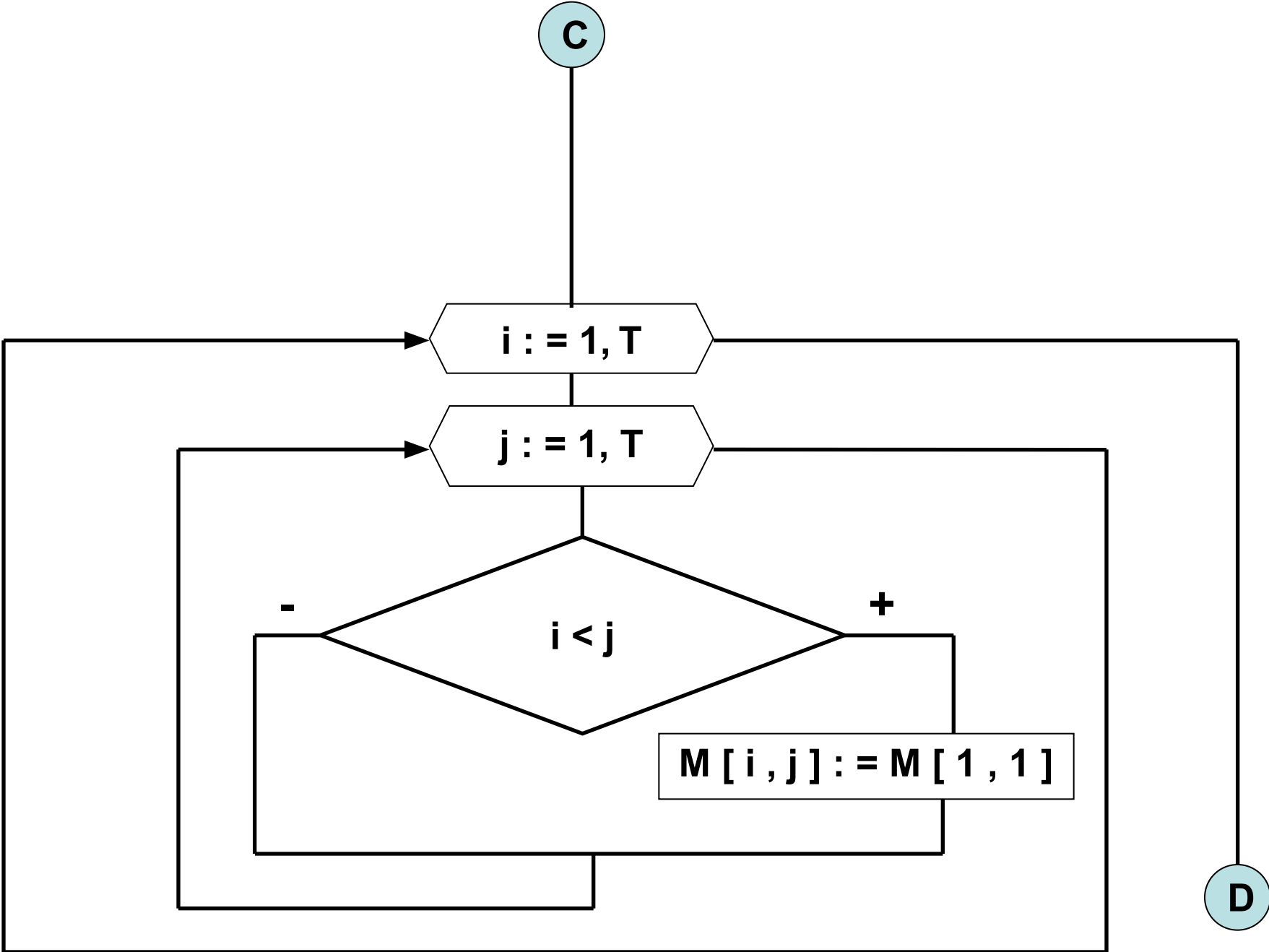
$i := 1, T$

$j := 1, T$

$i < j$

$M[i, j] := M[1, 1]$

D



D

Вывод сообщения

*Матрица
ПОСЛЕ изменений*

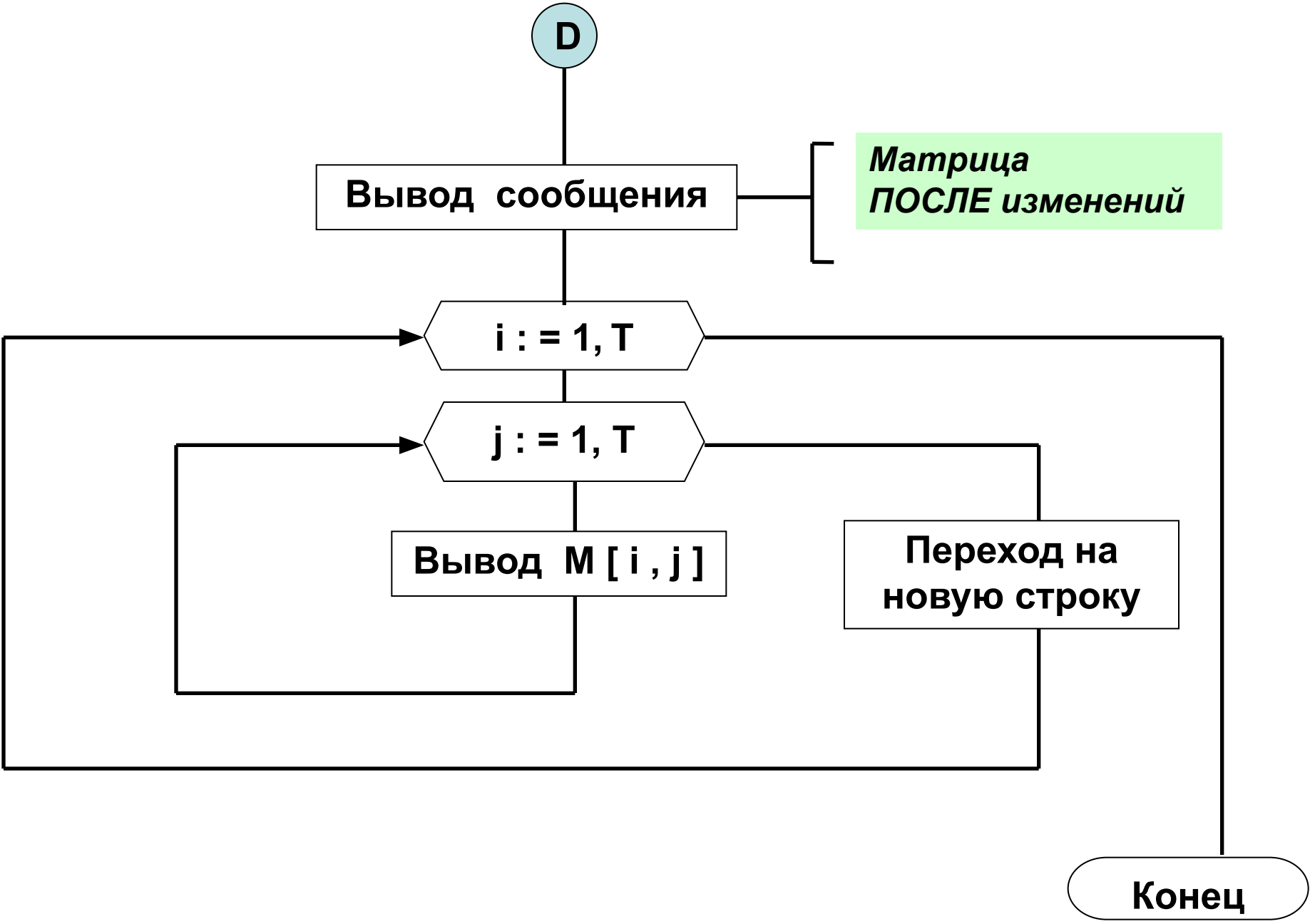
$i := 1, T$

$j := 1, T$

Вывод $M[i, j]$

Переход на
новую строку

Конец



Program Matrica ;

Var

M : array [1 .. 10 , 1 .. 10] of integer ; *{ матрица }*

i , j : byte ; *{ индексы элементов матрицы }*

T : integer ; *{ размерность квадратной матрицы }*

BEGIN

REPEAT *{ проверка корректности ввода размерности }*

WriteLn (' Введите размерность квадратной матрицы ') ;

ReadLn (T) ;

IF (T < 2) OR (T > 10) THEN

WriteLn (' Размерность введена НЕ верно... ') ;

UNTIL (T > = 2) and (T < = 10) ;

Randomize ; { инициализация генератора случайных чисел }

FOR i := 1 **to** T **do** { заполнение матрицы }

FOR j := 1 **to** T **do**

M [i , j] := **Random** (99) ;

WriteLn (' Матрица ДО изменения ') ;

FOR i := 1 **to** T **do** { вывод матрицы на экран }

begin

FOR j := 1 **to** T **do**

Write (M [i , j] : 6) ; { форматный вывод }

WriteLn ; { переход на новую строку }

end ;

```
FOR i := 1 to T do { замена элементов }  
FOR j := 1 to T do  
IF i < j THEN M [ i , j ] := M [ 1 , 1 ] ;
```

```
WriteLn ( ' Матрица ПОСЛЕ изменений ' ) ;
```

```
FOR i := 1 to T do { вывод матрицы на экран }  
begin  
FOR j := 1 to T do  
Write ( M [ i , j ] : 6 ) ;  
WriteLn ;  
end ;
```

```
ReadLn ;  
END .
```

The end ...