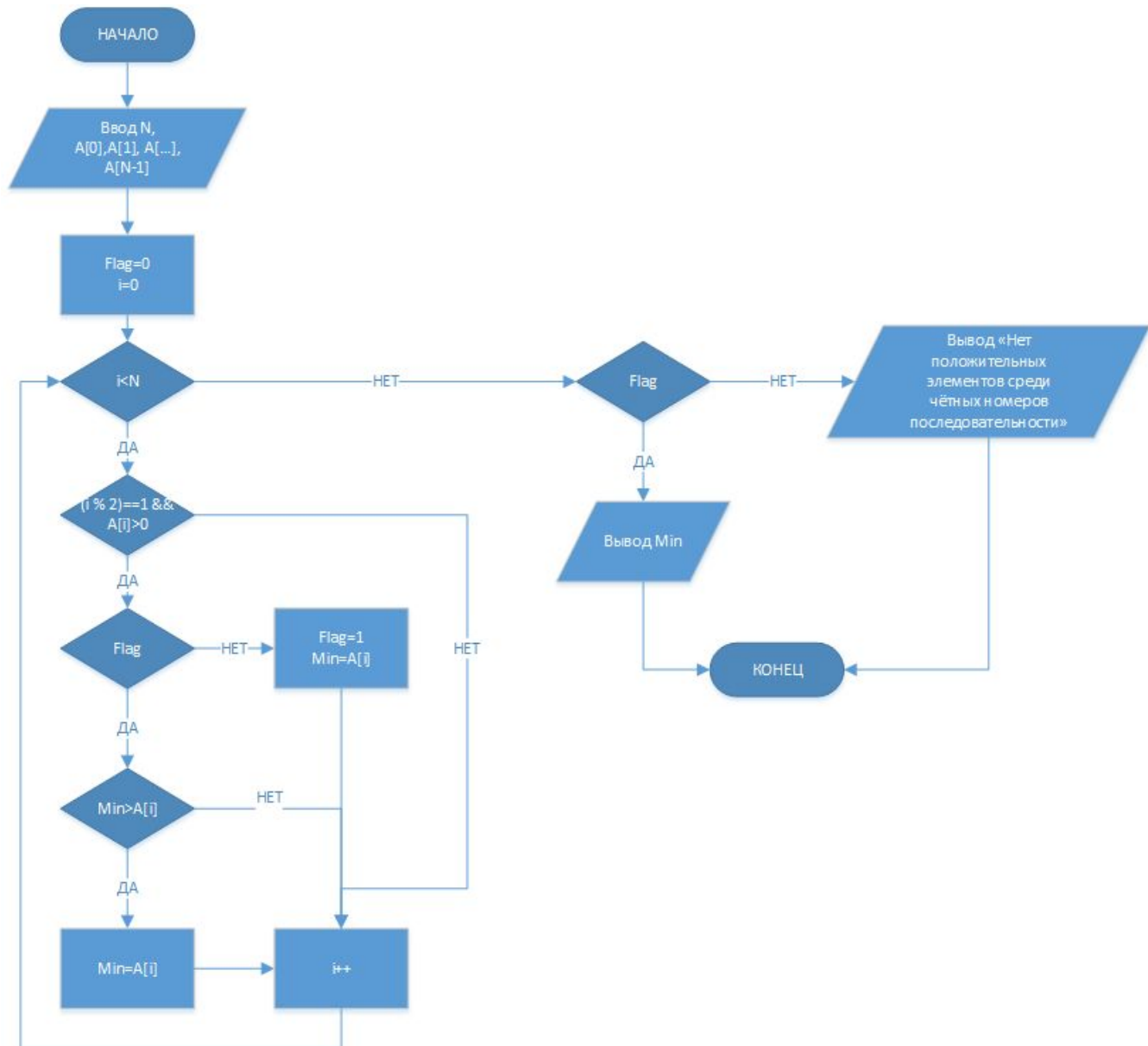
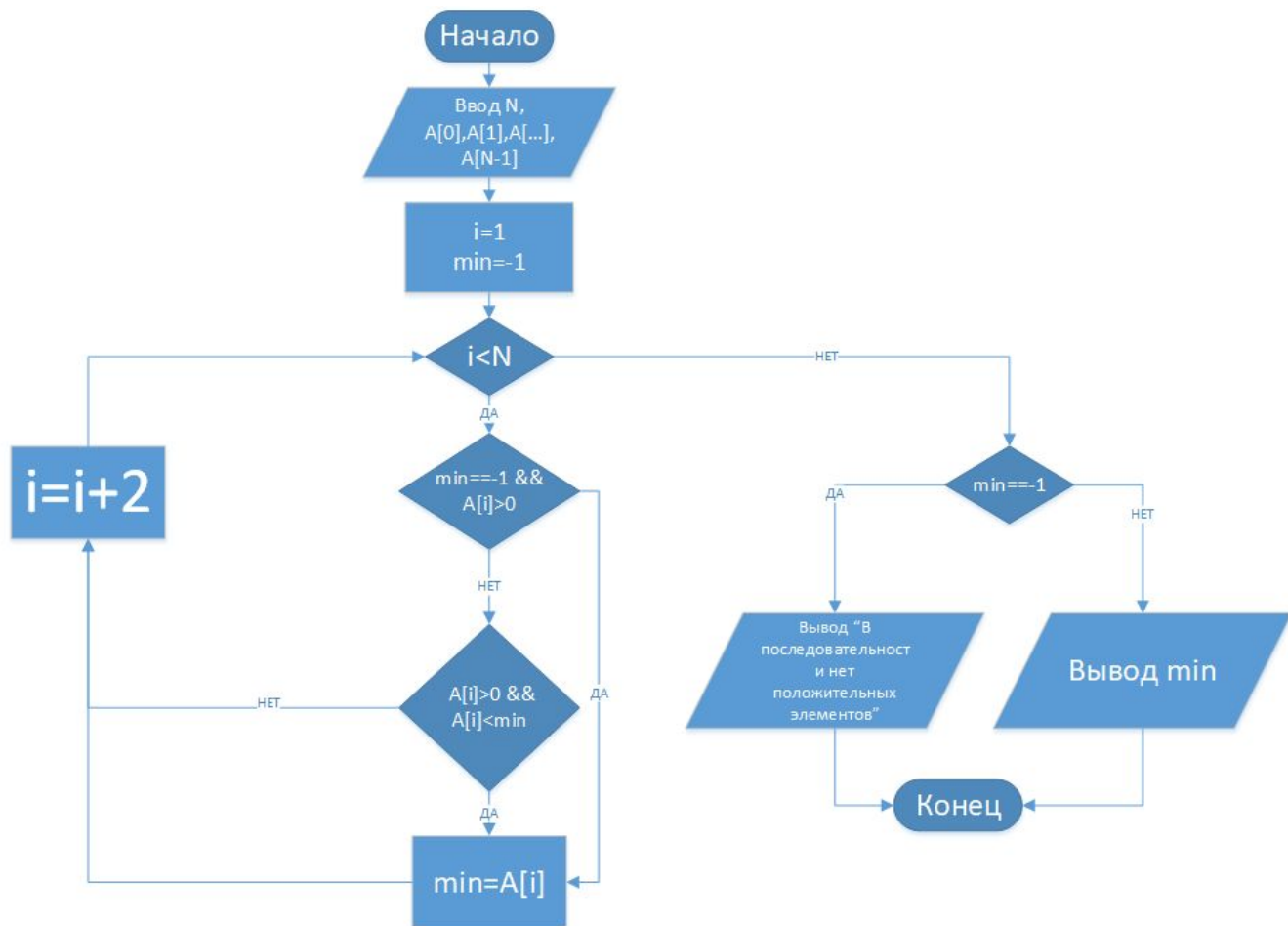


Типовые задания на составление блок-схем

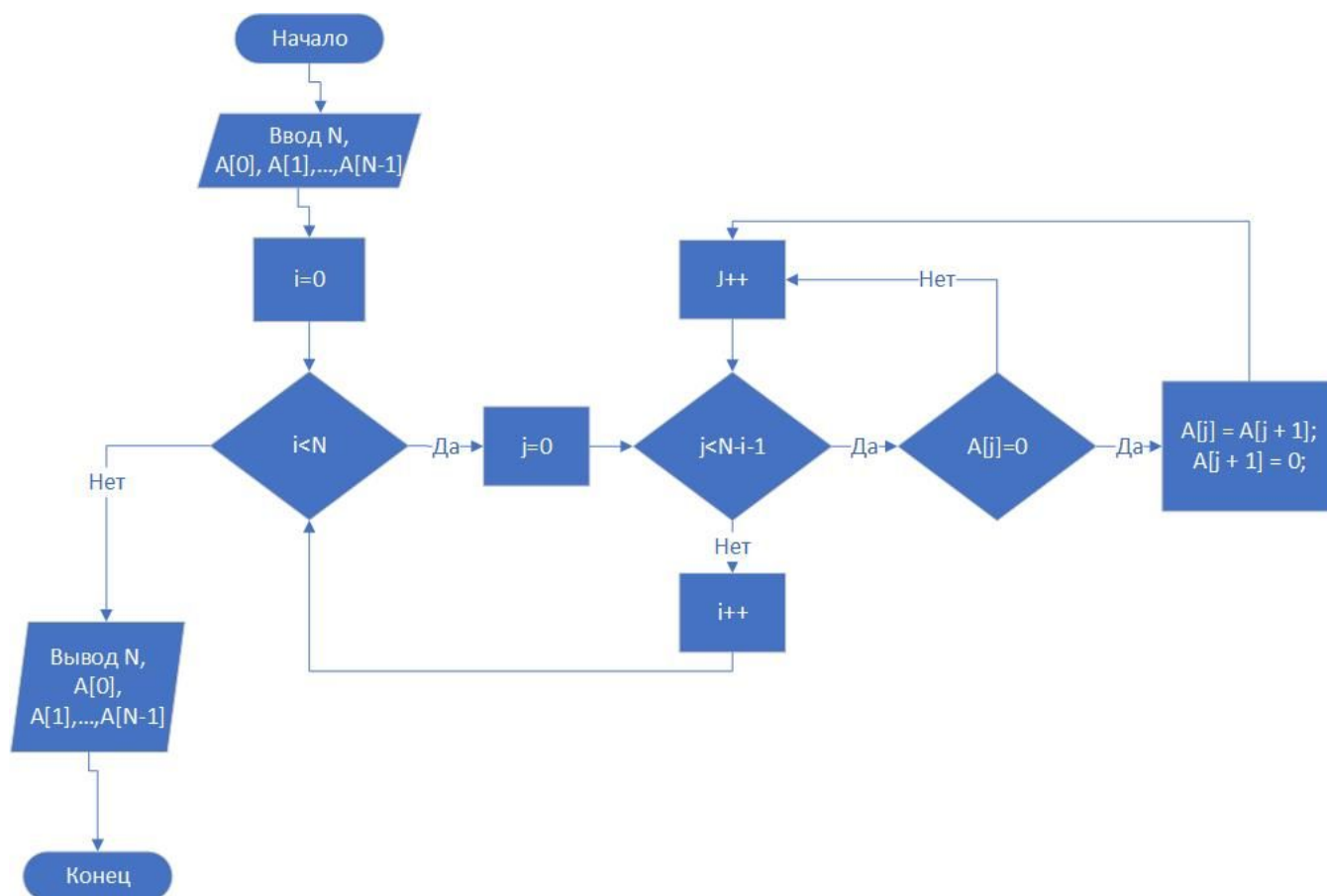
1. Для произвольной последовательности целых чисел размера N найти наименьший положительный элемент среди чётных номеров последовательности.



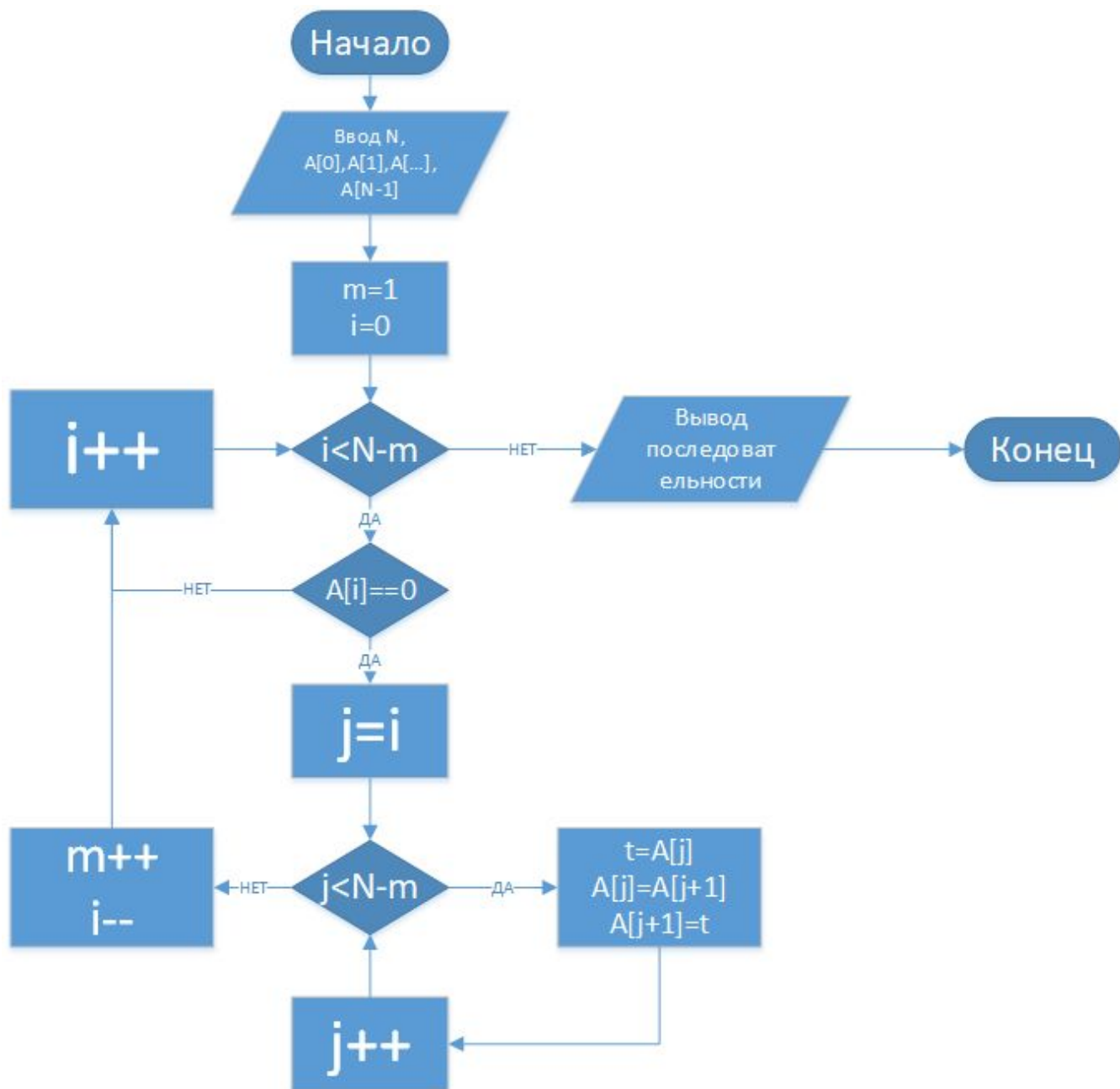
НОВЫЙ ВАРИК



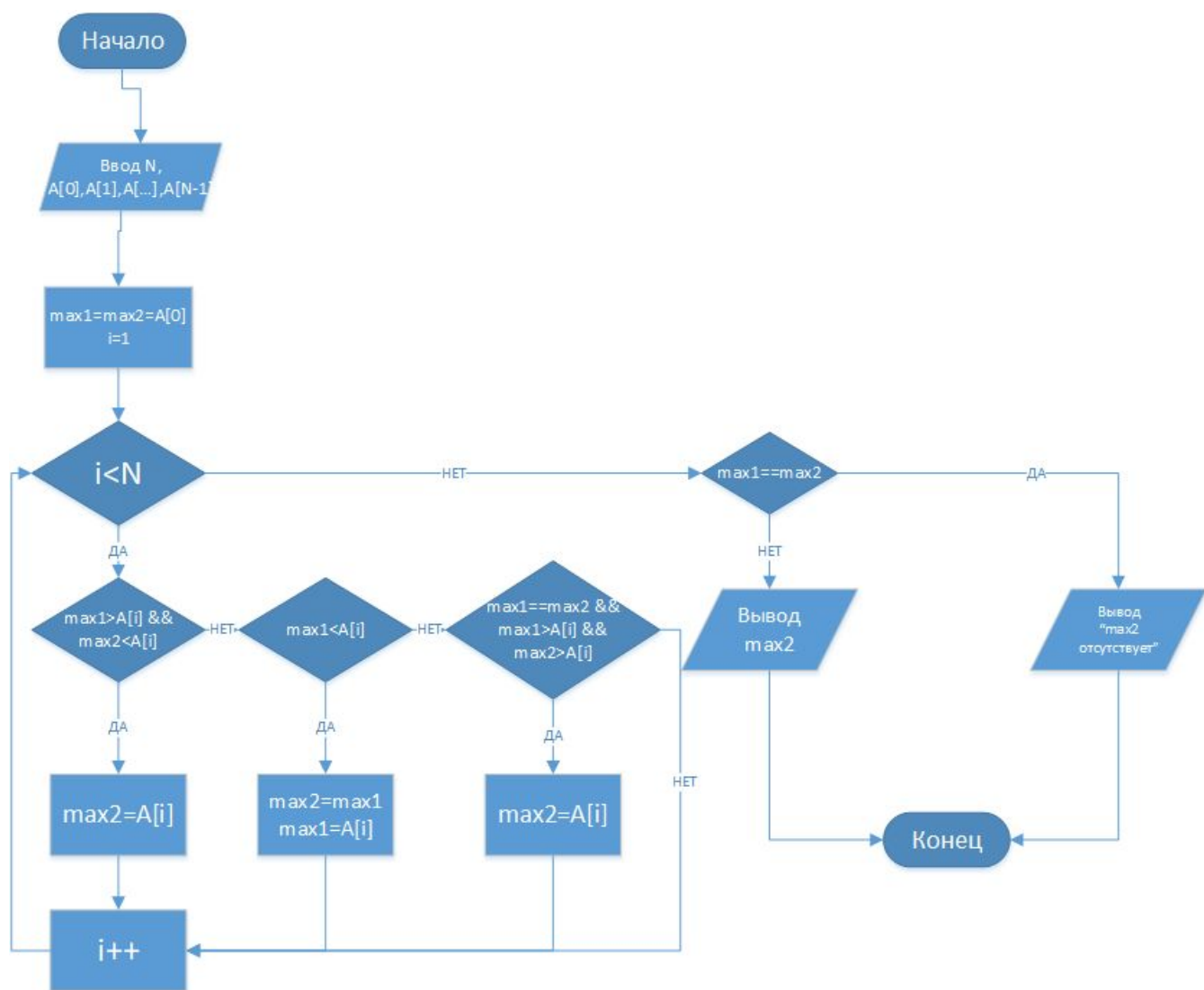
2. Дана последовательность целых чисел. Передвинуть все нули в конец последовательности. Порядок остальных элементов должен остаться тем же. Нельзя использовать дополнительную последовательность.



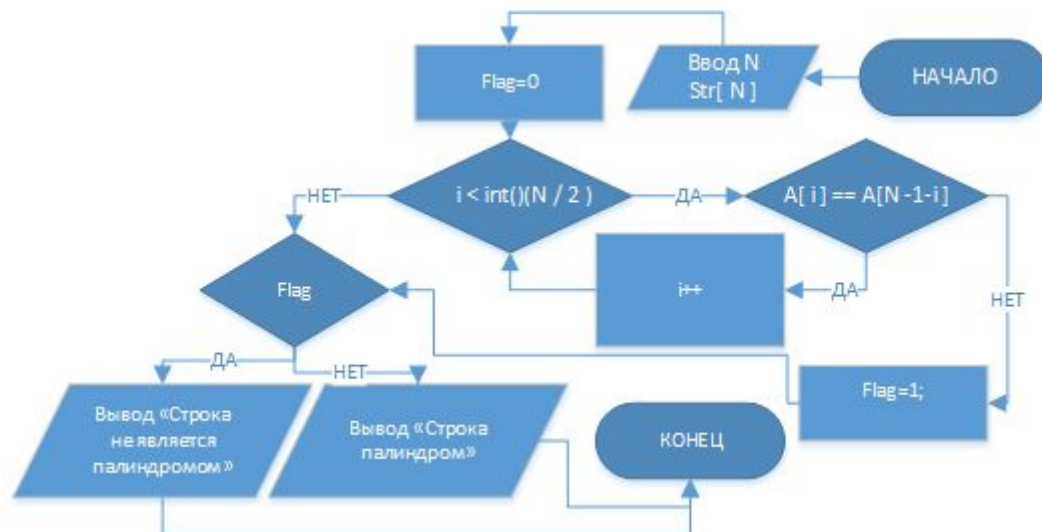
Свою версию подогнал, чекните плз



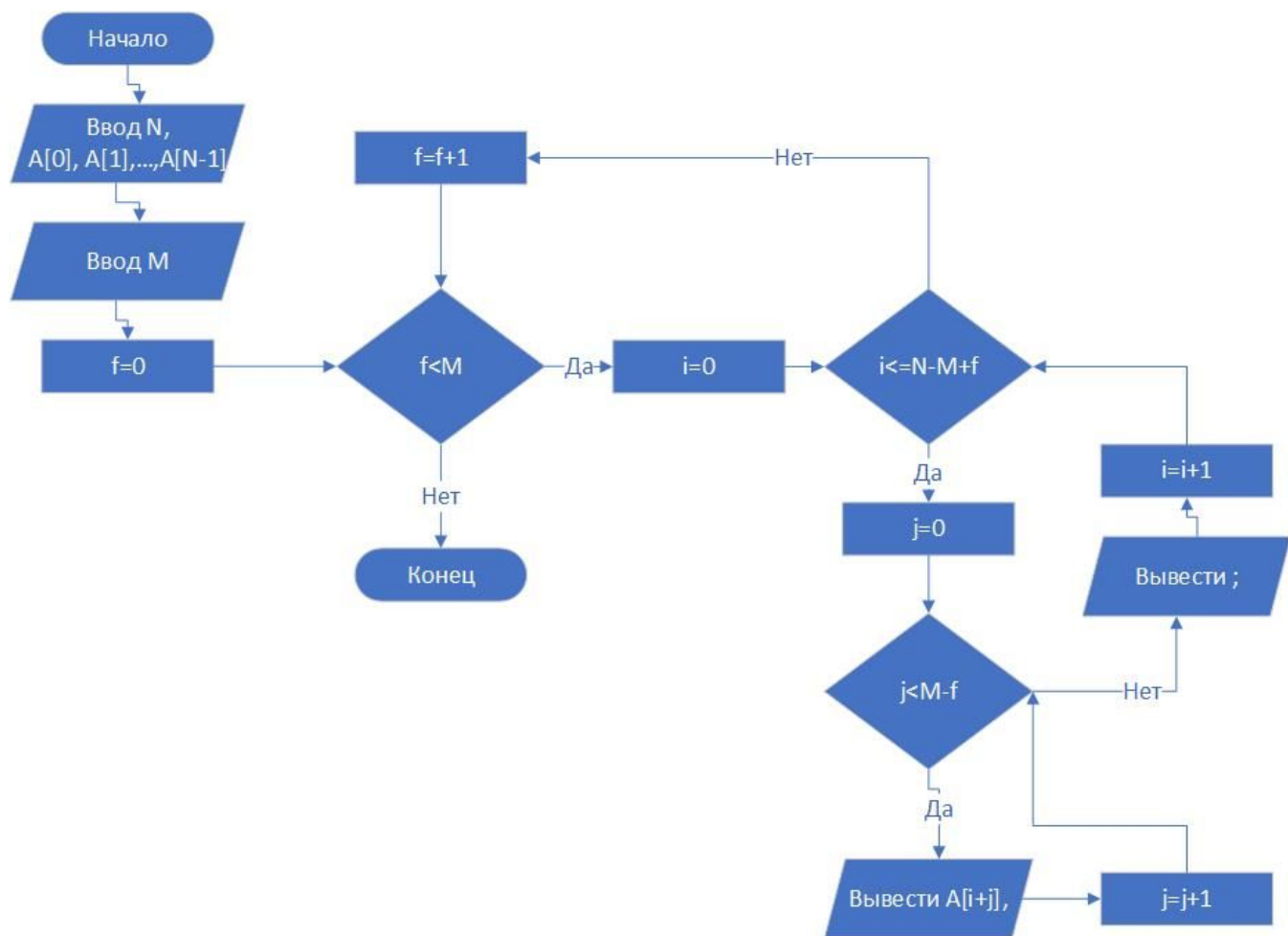
3. Дана последовательность целых чисел. Найти второй по величине элемент последовательности за один проход.



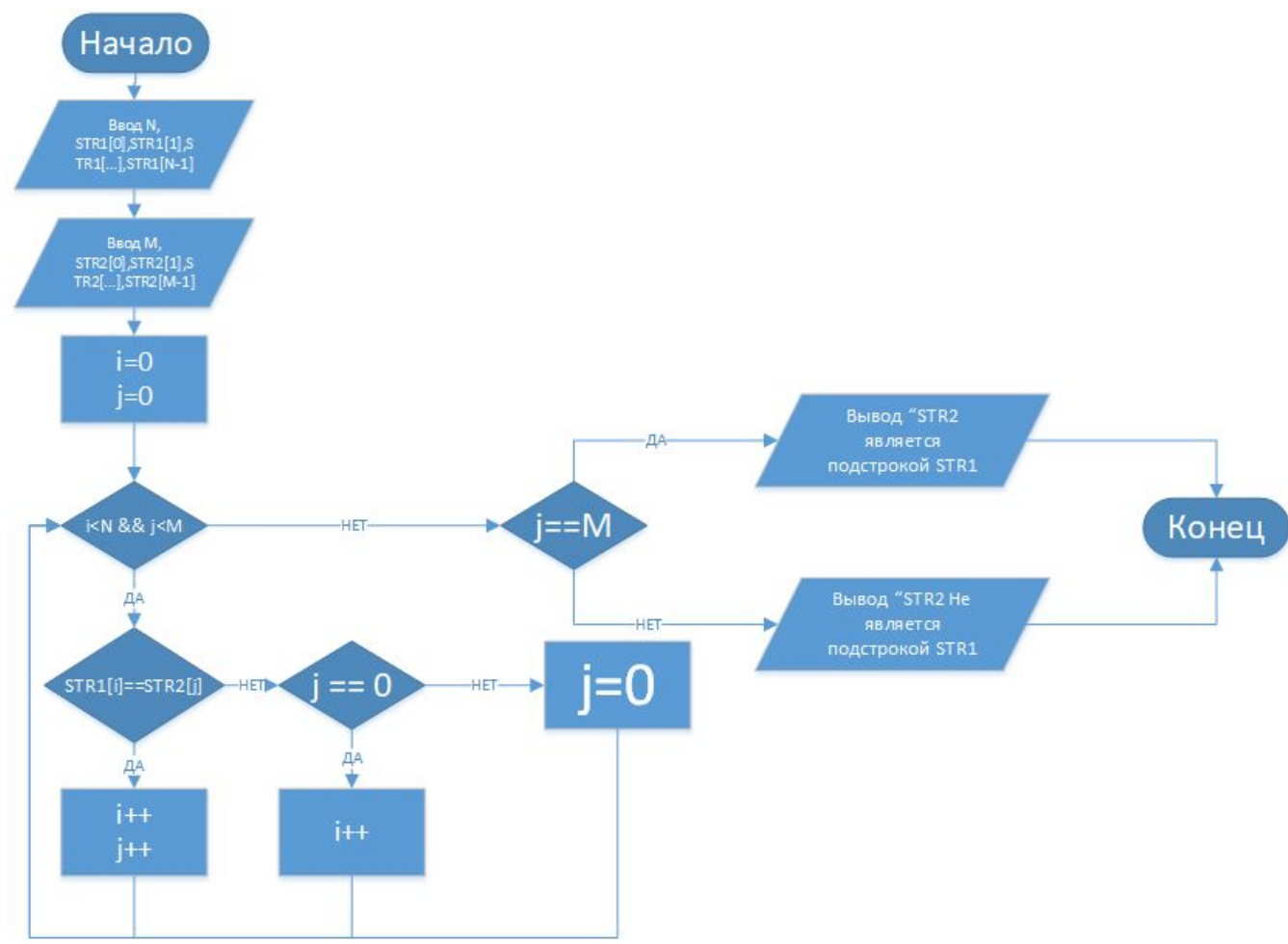
4. Проверить является ли строка палиндромом.



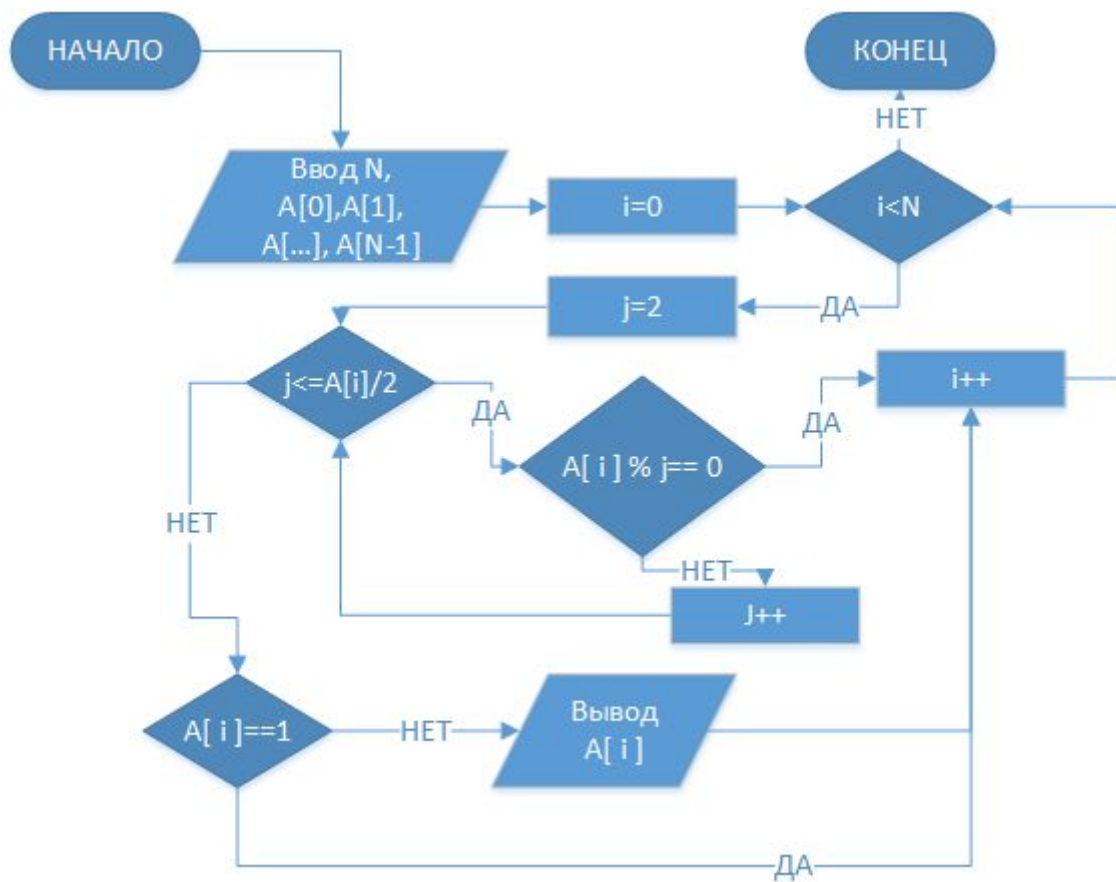
5. Дана строка размера N. Вывести все подстроки длиной не больше M.



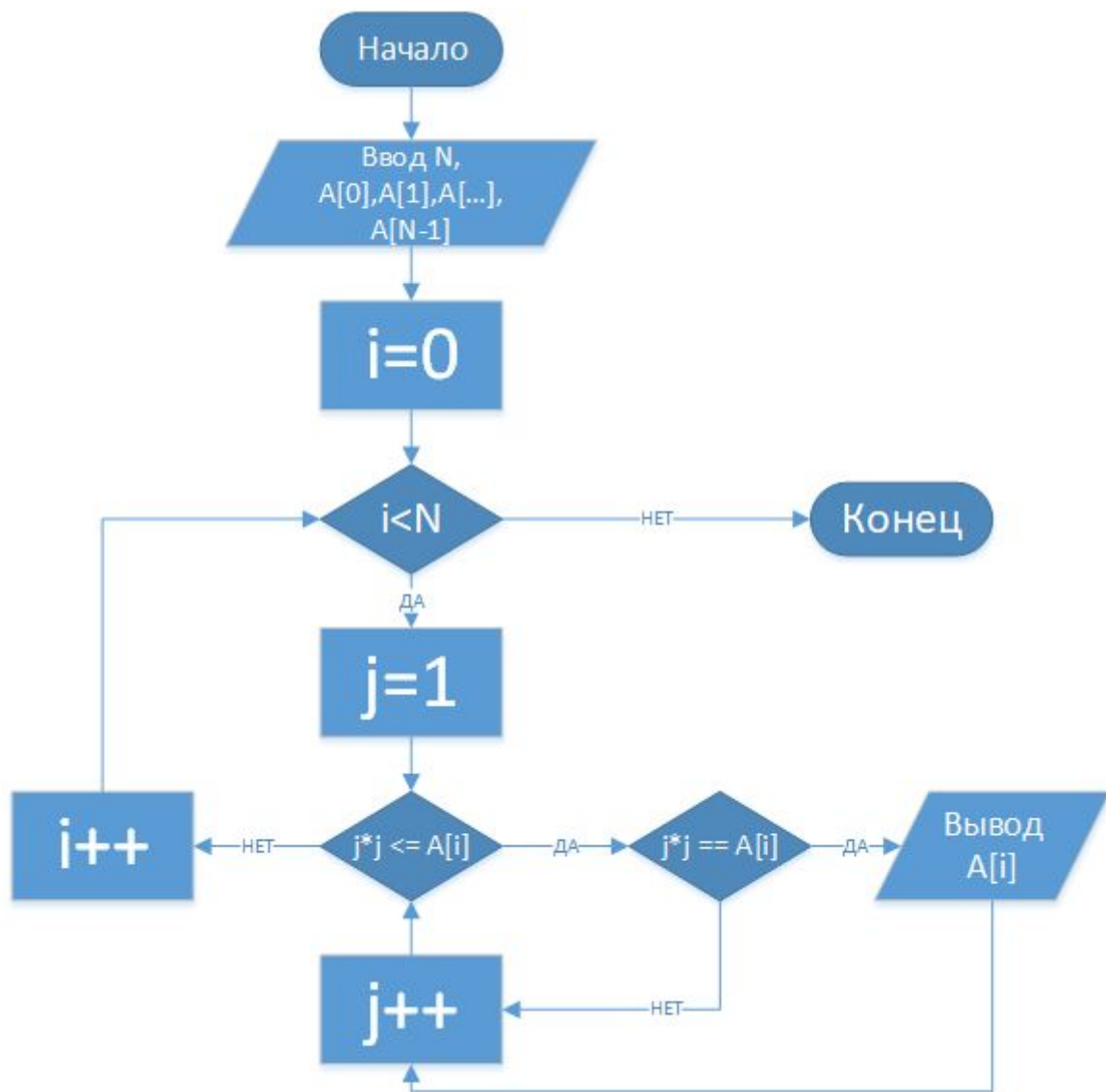
6. Даны строки STR1 и STR2, длины N и M соответственно. Проверить, является ли STR2 подстрокой STR1.



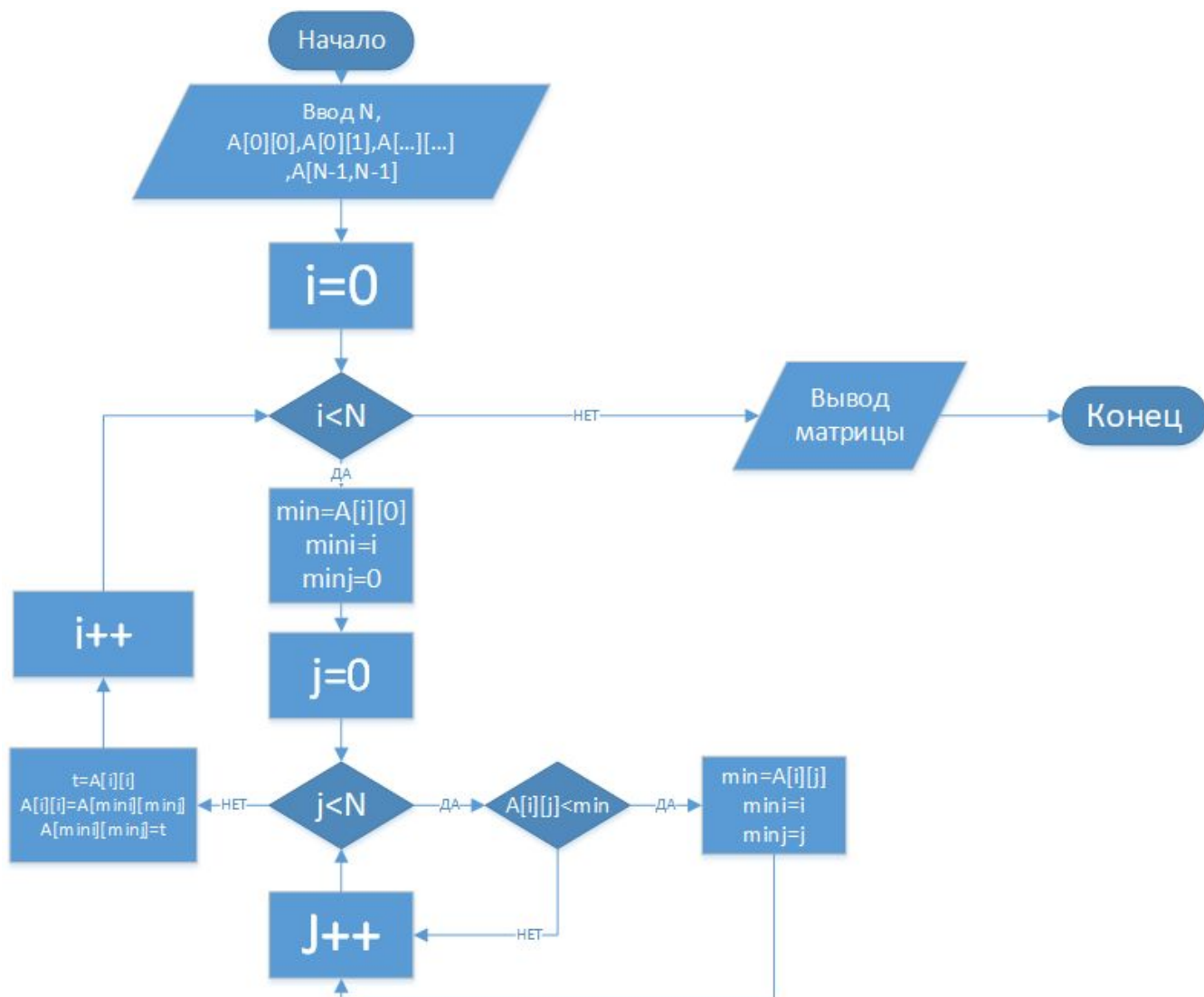
7. Дана последовательность размером N из натуральных чисел. Вывести только простые числа.



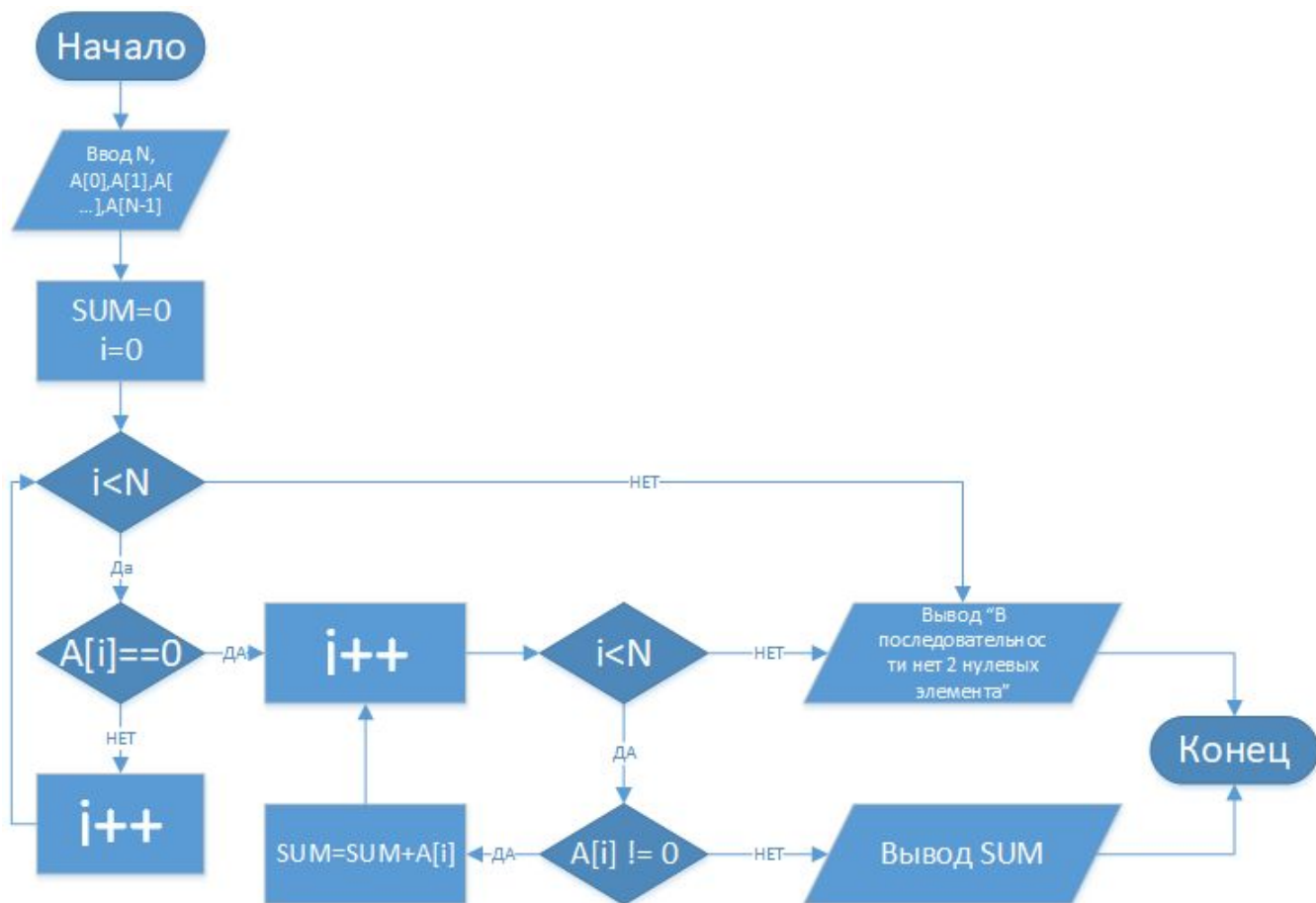
8. Дана последовательность натуральных чисел. Вывести числа, которые являются полными квадратами.



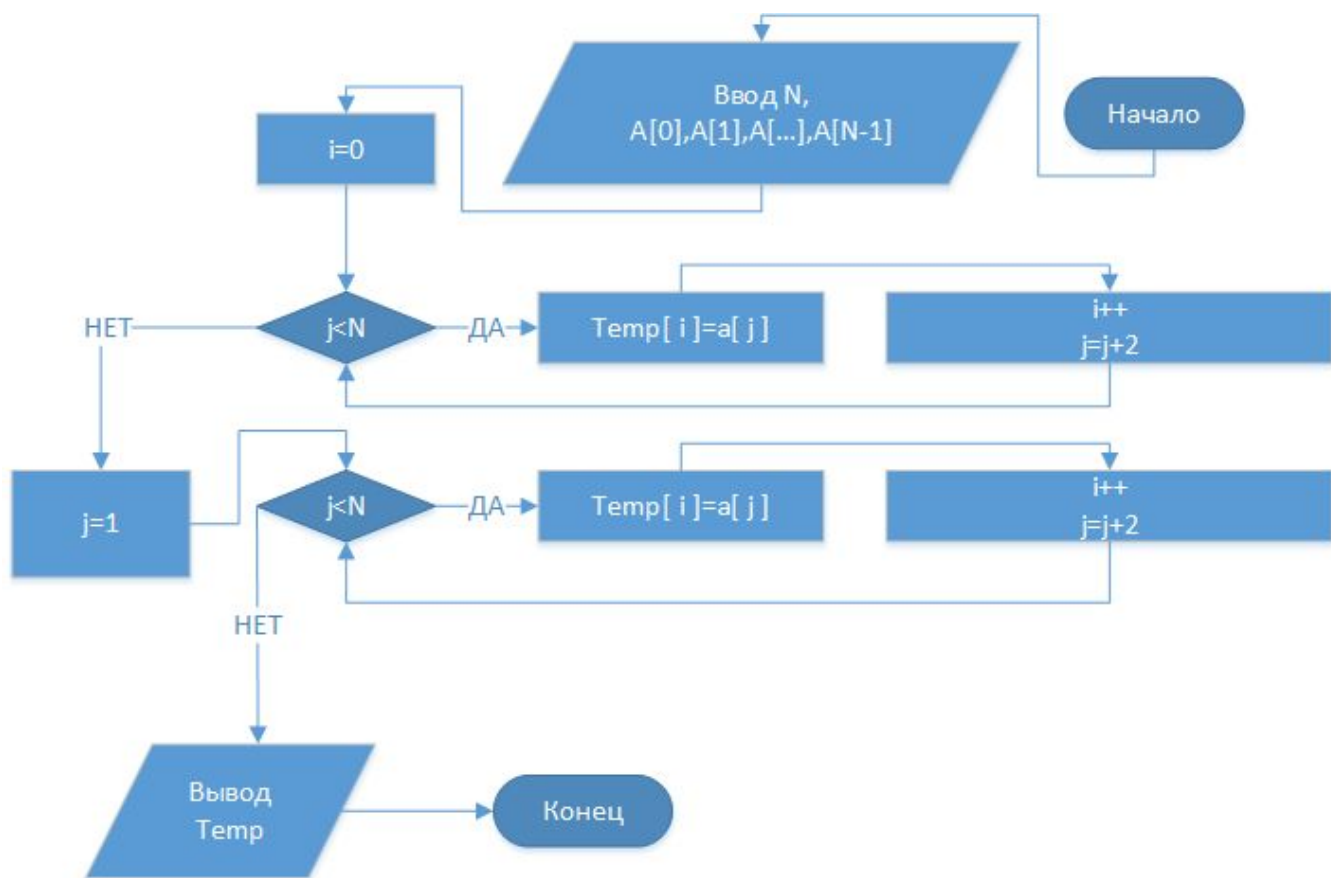
9. Дана квадратная матрица. Найти в каждой строке минимальный элемент и переставить его местами с элементом на главной диагонали.



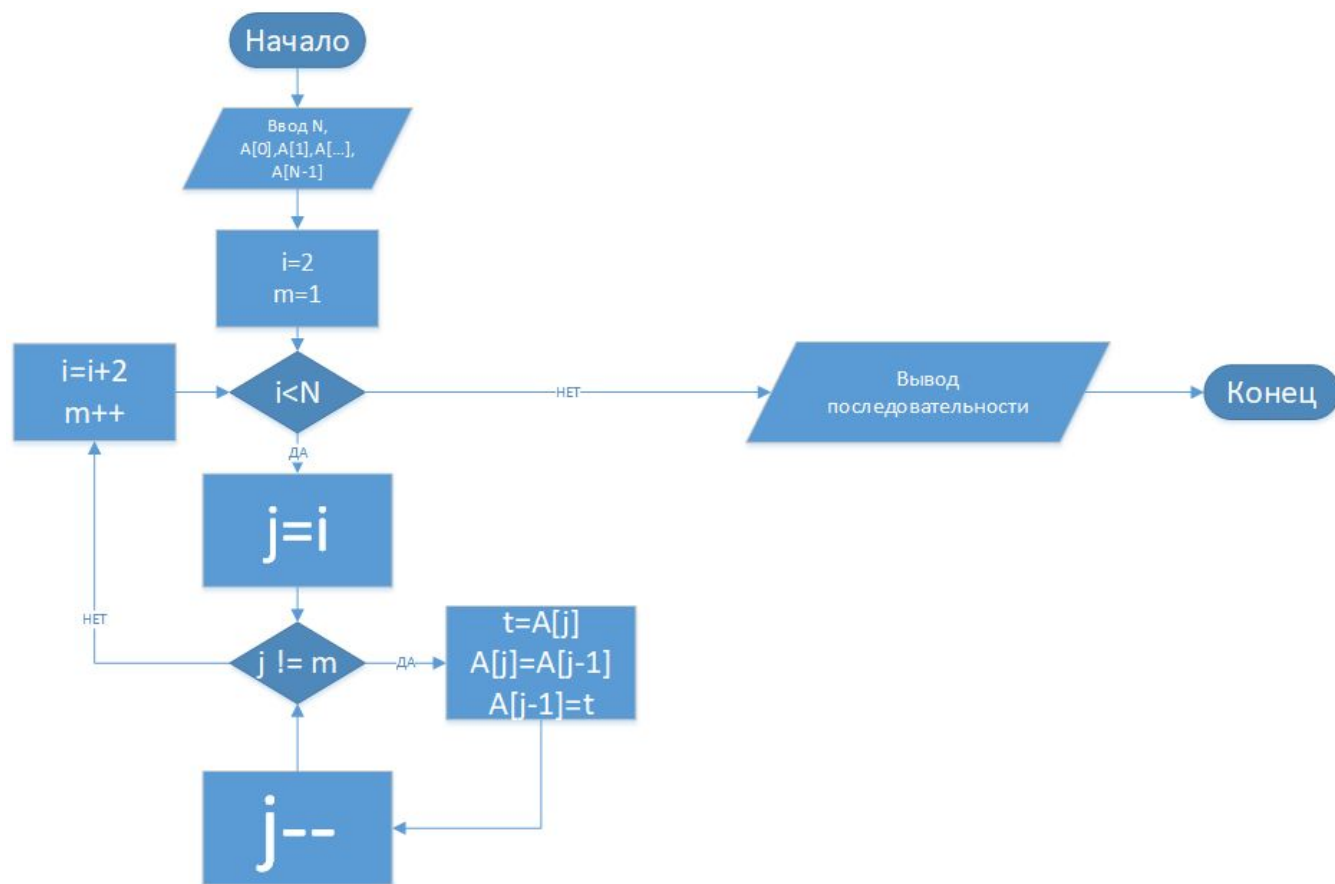
10. Дана последовательность целых чисел. Вычислить сумму элементов между двумя первыми нулевыми элементами.



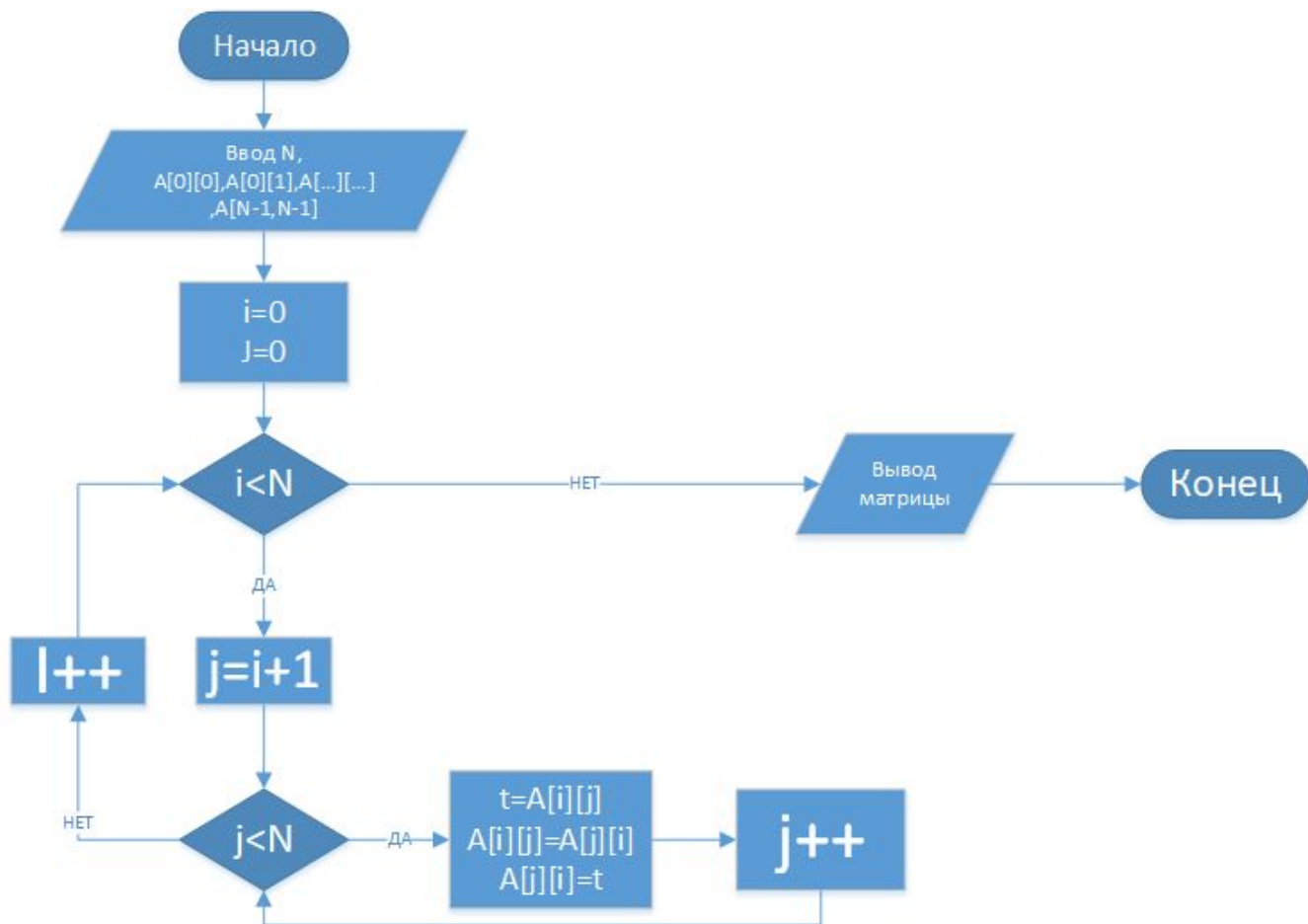
11. Дана последовательность целых чисел. Преобразовать её так чтобы в левой половине стояли числа, которые сначала стояли в нечётных позициях, в правой половине те, которые стояли в чётных позициях.



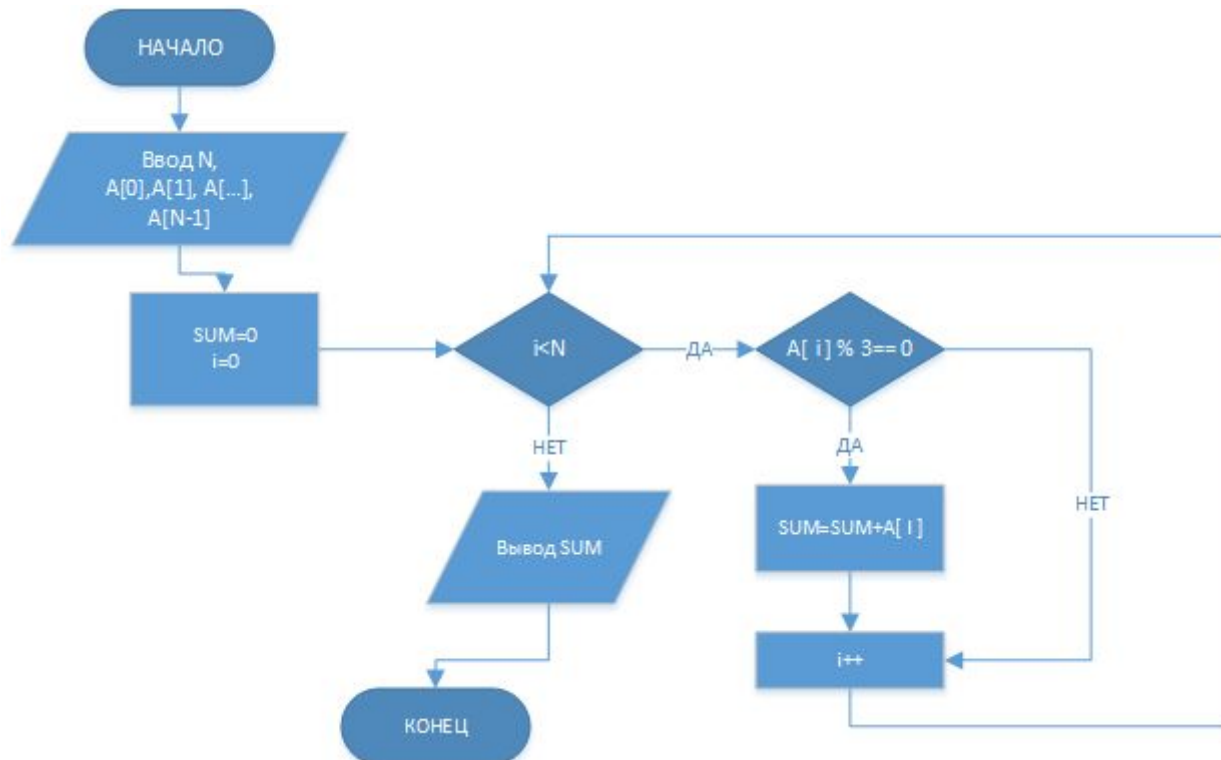
Новый вариант



12. Составить алгоритм транспонирования квадратной матрицы.



13. Дана последовательность целых чисел. Сложить элементы кратные трём.



Задания на тему «Машина Тьюринга» на оценку «хорошо» Для всех задач произвольные входные данные, но соответствующие начальному условию (числа могут быть любой разрядности, но числа обязательно целые или натуральные, их знак указан в задании, длина входной строки произвольная, но состоит только из нулей и единиц).

(morphett.info/turing/turing.html)

1. Сложить два натуральных числа

```
0 * * r 0
0 _ _ r test
test 0 0 r test
test _ _ l clear
test * * r q1
q1 * * r q1
q1 _ _ l q2
q2 0 9 l q2
q2 1 0 l q3
q2 2 1 l q3
q2 3 2 l q3
q2 4 3 l q3
q2 5 4 l q3
q2 6 5 l q3
q2 7 6 l q3
q2 8 7 l q3
q2 9 8 l q3
q3 * * l q3
q3 _ _ l q4
q4 0 1 r 0
q4 1 2 r 0
q4 2 3 r 0
q4 3 4 r 0
q4 4 5 r 0
q4 5 6 r 0
q4 6 7 r 0
q4 7 8 r 0
q4 8 9 r 0
q4 9 0 l q4
q4 _ 1 r 0
clear 0 _ l clear
clear _ _ * halt
```

2.(3) Сложить числа a и b , где a – положительное, b – отрицательное. Число a на ленте идёт обязательно перед числом b . (это и 2 и 3 и 4 все зависит от знака который ты поставишь перед числом)

0 - - r m1

0 * * r p1

p1 * * r p1

p1 _ _ r pt

pt - - r pm1

pt * * r pp1

pp1 * * r pp1

pp1 _ _ l pp2

pp2 1 0 l pp3

pp2 2 1 l pp3

pp2 3 2 l pp3

pp2 4 3 l pp3

pp2 5 4 l pp3

pp2 6 5 l pp3

pp2 7 6 l pp3

pp2 8 7 l pp3

pp2 9 8 l pp3

pp2 0 9 l pp2

pp3 * * l pp3

pp3 _ _ l pp4

pp4 0 1 r pp5

pp4 1 2 r pp5

pp4 2 3 r pp5

pp4 3 4 r pp5

pp4 4 5 r pp5

pp4 5 6 r pp5

pp4 6 7 r pp5

pp4 7 8 r pp5

pp4 8 9 r pp5

pp4 9 0 l pp4

pp4 _ 1 r pp5

pp5 * * r pp5

pp5 _ _ r ppt

ppt _ _ l ppc

ppt * * r pp1

ppc 0 _ l ppc

ppc _ _ * halt

pm0 * * r pm0

pm0 _ _ r test

test - - r test

test 0 0 r test

test _ _ l clear

test * * r pm1

pm1 * * r pm1
pm1 _ _ l q2
q2 0 9 l q2
q2 1 0 l q3
q2 2 1 l q3
q2 3 2 l q3
q2 4 3 l q3
q2 5 4 l q3
q2 6 5 l q3
q2 7 6 l q3
q2 8 7 l q3
q2 9 8 l q3
q3 - - l q3
q3 * * l q3
q3 _ _ l test2
q4 1 0 r pm0
q4 2 1 r pm0
q4 3 2 r pm0
q4 4 3 r pm0
q4 5 4 r pm0
q4 6 5 r pm0
q4 7 6 r pm0
q4 8 7 r pm0
q4 9 8 r pm0
q4 0 9 l q4
q41 * * r q41
q41 _ _ l q4
test2 0 0 l test2
test2 _ - r q6
test2 * * r q41
q6 * * r q6
q6 _ _ l q7
q7 0 1 l 01
q7 1 2 l 01
q7 2 3 l 01
q7 3 4 l 01
q7 4 5 l 01
q7 5 6 l 01
q7 6 7 l 01
q7 7 8 l 01
q7 8 9 l 01
q7 9 0 r q7
01 * * r 01
01 _ _ r test3
test3 - - r test3
test3 0 0 r test3

test3 __ l clear
test3 ** r q8
q8 ** r q8
q8 __ l q9
q9 0 9 l q9
q9 1 0 l q10
q9 2 1 l q10
q9 3 2 l q10
q9 4 3 l q10
q9 5 4 l q10
q9 6 5 l q10
q9 7 6 l q10
q9 8 7 l q10
q9 9 8 l q10
q10 ** l q10
q10 __ l q11
q11 0 1 r 01
q11 1 2 r 01
q11 2 3 r 01
q11 3 4 r 01
q11 4 5 r 01
q11 5 6 r 01
q11 6 7 r 01
q11 7 8 r 01
q11 8 9 r 01
q11 9 0 l q11
q11 - 1 l minus
minus _ - r 01
clear 0 _ l clear
clear - _ l clear
clear __ * halt
m1 ** r m1
m1 __ r mt
mt - - r mm1
mt ** r mp1
mp1 ** r mp1
mp1 __ l mp2
mp2 1 0 l mp3
mp2 2 1 l mp3
mp2 3 2 l mp3
mp2 4 3 l mp3
mp2 5 4 l mp3
mp2 6 5 l mp3
mp2 7 6 l mp3
mp2 8 7 l mp3
mp2 9 8 l mp3

mp2 0 9 l mp2
mp3 * * l mp3
mp3 _ _ l mpte
mpte 0 0 l mpte
mpte - _ r mpt1
mpte * * r mp4
mp4 * * r mp4
mp4 _ _ l mp5
mp5 1 0 r mp6
mp5 2 1 r mp6
mp5 3 2 r mp6
mp5 4 3 r mp6
mp5 5 4 r mp6
mp5 6 5 r mp6
mp5 7 6 r mp6
mp5 8 7 r mp6
mp5 9 8 r mp6
mp5 0 9 l mp5
mp6 * * r mp6
mp6 _ _ r mpe
mpe 0 0 r mpe
mpe _ _ l mpec
mpe * * r mp1
mpec 0 _ l mpec
mpec _ _ l mpecl
mpecl 0 0 l mpecl
mpecl - _ * halt
mpecl * * * halt
mpt1 0 0 r mpt1
mpt1 _ _ l mpt2
mpt2 0 1 r mpt3
mpt2 1 2 r mpt3
mpt2 2 3 r mpt3
mpt2 3 4 r mpt3
mpt2 4 5 r mpt3
mpt2 5 6 r mpt3
mpt2 6 7 r mpt3
mpt2 7 8 r mpt3
mpt2 8 9 r mpt3
mpt2 9 0 l mpt2
mpt2 _ 1 r mpt3
mpt3 * * r mpt3
mpt3 _ _ r mptt
mptt 0 0 r mptt
mptt _ _ l mpttc
mptt * * r mpt4

mpt4 * * r mpt4
mpt4 _ _ l mpt5
mpt5 1 0 l mpt6
mpt5 2 1 l mpt6
mpt5 3 2 l mpt6
mpt5 4 3 l mpt6
mpt5 5 4 l mpt6
mpt5 6 5 l mpt6
mpt5 7 6 l mpt6
mpt5 8 7 l mpt6
mpt5 9 8 l mpt6
mpt5 0 9 l mpt5
mpt6 * * l mpt6
mpt6 _ _ l mpt2
mpttc 0 _ l mpttc
mpttc _ _ * halt
mm1 * * r mm1
mm1 _ _ l mm2
mm2 1 0 l mm3
mm2 2 1 l mm3
mm2 3 2 l mm3
mm2 4 3 l mm3
mm2 5 4 l mm3
mm2 6 5 l mm3
mm2 7 6 l mm3
mm2 8 7 l mm3
mm2 9 8 l mm3
mm2 0 9 l mm2
mm3 * * l mm3
mm3 _ _ l mm4
mm4 0 1 r mm5
mm4 1 2 r mm5
mm4 2 3 r mm5
mm4 3 4 r mm5
mm4 4 5 r mm5
mm4 5 6 r mm5
mm4 6 7 r mm5
mm4 7 8 r mm5
mm4 8 9 r mm5
mm4 9 0 l mm4
mm4 - 1 l min
min _ - r mm5
mm5 * * r mm5
mm5 _ _ r mmt
mmt - - r mmt
mmt 0 0 r mmt

```

mmt _ _ l mmc
mmt * * r mm1
mmc 0 _ l mmc
mmc - _ * halt

```

(3). Сложить числа a и b , где a – отрицательное, b – положительное. Число a на ленте идёт обязательно перед числом b .

(4). Сложить два отрицательных числа.

5. Дана последовательность из нулей и единиц. Вывести ноль, если количество нулей больше чем единиц. Вывести единицу, если количество единиц больше чем нулей. Иначе вывести двойку

```

0 0 0 r ta
0 1 1 r tb
ta 0 0 r ta
ta _ _ l clsa
ta 1 1 l ta1
clsa 0 _ l clsa
clsa _ 0 * halt
ta1 0 0 l ta1
ta1 _ _ r prov
tb 1 1 r tb
tb 0 0 l tb1
tb _ _ l clsb
clsb 1 _ l clsb
clsb _ 1 * halt
tb1 1 1 l tb1
tb1 _ _ r prov
prov 0 0 * a0
prov 1 1 * b0
prov _ _ r q1
a0 0 _ r a1
b0 1 _ r b1
a1 0 0 r a1
a1 1 1 r a1
a1 _ _ r a2
a2 _ 1 l an1
a2 * * r ap1
a2 - - r amte
an1 * * l an1
an1 _ _ l an2
an2 0 0 l an2
an2 1 1 l an2
an2 _ _ r prov
ap1 * * r ap1
ap1 _ _ l ap2
ap2 0 1 l an1

```

ap2 1 2 | an1
ap2 2 3 | an1
ap2 3 4 | an1
ap2 4 5 | an1
ap2 5 6 | an1
ap2 6 7 | an1
ap2 7 8 | an1
ap2 8 9 | an1
ap2 9 0 | ap2
ap2 __ r aw1
aw1 0 0 r aw1
aw1 __ l aw2
aw2 0 _ r aw0
aw2 __ l aw3
aw0 _ 0 l aw2
aw3 __ r aw4
aw4 _ 1 l an1
aw3 0 0 * aw2
amte 0 0 r amte
amte __ l amc
amte * * r am1
amc 0 _ l amc
amc - 1 l an1
am1 * * r am1
am1 __ l am2
am2 1 0 l an1
am2 2 1 l an1
am2 3 2 l an1
am2 4 3 l an1
am2 5 4 l an1
am2 6 5 l an1
am2 7 6 l an1
am2 8 7 l an1
am2 9 8 l an1
am2 0 9 l am2
b1 0 0 r b1
b1 1 1 r b1
b1 __ r b2
b2 _ - r m1
m1 _ 1 l an1
b2 - - r bm1
b2 * * * bpt1
bm1 * * r bm1
bm1 __ l bm2
bm2 0 1 l an1
bm2 1 2 l an1

bm2 2 3 l an1
bm2 3 4 l an1
bm2 4 5 l an1
bm2 5 6 l an1
bm2 6 7 l an1
bm2 7 8 l an1
bm2 8 9 l an1
bm2 9 0 l bm2
bm2 - - r bmr1
bmr1 0 0 r bmr1
bmr1 _ _ l bmr2
bmr2 0 _ r bmr0
bmr2 _ _ l bmr3
bmr0 _ 0 l bmr2
bmr3 0 0 * bmr1
bmr3 - - r bmr4
bmr4 _ 1 l an1
bpt1 0 0 r bpt1
bpt1 _ _ l bptc
bpt1 * * r bp1
bptc 0 _ l bptc
bptc _ _ r m2
m2 _ - r m3
m3 _ 1 l an1
bp1 * * r bp1
bp1 _ _ l bp2
bp2 1 0 l an1
bp2 2 1 l an1
bp2 3 2 l an1
bp2 4 3 l an1
bp2 5 4 l an1
bp2 6 5 l an1
bp2 7 6 l an1
bp2 8 7 l an1
bp2 9 8 l an1
bp2 0 9 l bp2
q1 - - r qt
q1 0 0 r qt
q1 * * r q1c1
q1c1 * * r q1c1
q1c1 _ _ l q1c2
q1c2 * _ l q1c2
q1c2 _ 0 * halt
qt 0 0 r qt
qt * * r q2
q2 * * r q2

q2 _ _ l qc
qc * _ l qc
qc _ 0 * halt
qc - 1 * halt
qt _ _ l qtc
qtc 0 _ l qtc
qtc _ 2 * halt
qtc - 2 * halt