

**Материалы заданий заключительного этапа
«Открытой олимпиады школьников «Информационные технологии»
(№ 57 Перечня олимпиад школьников)**

Задания заключительного этапа. 11 класс.

1. Информация и её кодирование (1 балл)

Вариант 1 Ответ: C0

Сколько существует таких натуральных чисел, что их запись в шестнадцатеричной системе счисления будет иметь ровно две значащих цифры, а в восьмеричной системе счисления – ровно три значащих цифры? Ответ запишите в шестнадцатеричной системе счисления.

Вариант 2 Ответ: E00

Сколько существует таких натуральных чисел, что их запись в шестнадцатеричной системе счисления будет иметь ровно три значащих цифры, а в восьмеричной системе счисления – ровно четыре значащих цифры? Ответ запишите в шестнадцатеричной системе счисления.

Вариант 3 Ответ: 7000

Сколько существует таких натуральных чисел, что их запись в шестнадцатеричной системе счисления будет иметь ровно четыре значащих цифры, а в восьмеричной системе счисления – ровно пять значащих цифр? Ответ запишите в шестнадцатеричной системе счисления.

Вариант 4 Ответ: 8000

Сколько существует таких натуральных чисел, что их запись в шестнадцатеричной системе счисления будет иметь ровно четыре значащих цифры, а в восьмеричной системе счисления – ровно шесть значащих цифр? Ответ запишите в шестнадцатеричной системе счисления.

Вариант 5 Ответ: C0000

Сколько существует таких натуральных чисел, что их запись в шестнадцатеричной системе счисления будет иметь ровно пять значащих цифр, а в восьмеричной системе счисления – ровно семь значащих цифр? Ответ запишите в шестнадцатеричной системе счисления.

Вариант 6 Ответ: E00000

Сколько существует таких натуральных чисел, что их запись в шестнадцатеричной системе счисления будет иметь ровно шесть значащих цифр, а в восьмеричной системе счисления – ровно восемь значащих цифр? Ответ запишите в шестнадцатеричной системе счисления.

2. Информация и её кодирование (1 балл)

Вариант 1 Ответ: GEJ

Код Хэмминга является примером кода с исправлением ошибок. Рассмотрим пример кодовой таблицы символов A, B, C, D, E, F, G, H, J и K, сопоставляющей каждому символу двоичную последовательность, построенную на основе кода Хэмминга.

A	0 0 0 0 0 0 0
B	1 1 0 1 0 0 1
C	0 1 0 1 0 1 0
D	1 0 0 0 0 1 1
E	1 0 0 1 1 0 0
F	0 1 0 0 1 0 1
G	1 1 0 0 1 1 0
H	0 0 0 1 1 1 1
J	1 1 1 0 0 0 0
K	0 0 1 1 0 0 1

Утверждается, что при кодировании символов с использованием данной кодовой таблицы появляется возможность однозначно декодировать значение символа по двоичной последовательности, в которой может быть допущена одна ошибка (один двоичный разряд изменил свое значение).

Были переданы три символа из указанного диапазона, закодированные с использованием кода приведенной кодовой таблицы. Известно, что при передаче кода каждого символа произошла ошибка в одном из двоичных разрядов. В результате было получено следующее сообщение:

1 0 0 0 1 1 0 1 0 1 1 1 0 0 0 1 1 0 0 0 0

Декодируйте полученное сообщение. В ответе укажите последовательность из трех символов в порядке их следования в сообщении без пробелов.

Вариант 2 Ответ: НВС

Код Хэмминга является примером кода с исправлением ошибок. Рассмотрим пример кодовой таблицы символов А, В, С, D, E, F, G, H, J и К, сопоставляющей каждому символу двоичную последовательность, построенную на основе кода Хэмминга.

А	0 0 0 0 0 0 0
В	1 1 0 1 0 0 1
С	0 1 0 1 0 1 0
D	1 0 0 0 0 1 1
E	1 0 0 1 1 0 0
F	0 1 0 0 1 0 1
G	1 1 0 0 1 1 0
H	0 0 0 1 1 1 1
J	1 1 1 0 0 0 0
К	0 0 1 1 0 0 1

Утверждается, что при кодировании символов с использованием данной кодовой таблицы появляется возможность однозначно декодировать значение символа по двоичной последовательности, в которой может быть допущена одна ошибка (один двоичный разряд изменил свое значение).

Были переданы три символа из указанного диапазона, закодированные с использованием кода приведенной кодовой таблицы. Известно, что при передаче кода каждого символа произошла ошибка в одном из двоичных разрядов. В результате было получено следующее сообщение:

0 1 0 1 1 1 1 1 1 1 0 0 1 0 1 0 1 1 1 0

Декодируйте полученное сообщение. В ответе укажите последовательность из трех символов в порядке их следования в сообщении без пробелов.

Вариант 3 Ответ: DKВ

Код Хэмминга является примером кода с исправлением ошибок. Рассмотрим пример кодовой таблицы символов от А, В, С, D, E, F, G, H, J и К, сопоставляющей каждому символу двоичную последовательность, построенную на основе кода Хэмминга.

А	0 0 0 0 0 0 0
В	1 1 0 1 0 0 1
С	0 1 0 1 0 1 0
D	1 0 0 0 0 1 1
E	1 0 0 1 1 0 0
F	0 1 0 0 1 0 1
G	1 1 0 0 1 1 0
H	0 0 0 1 1 1 1
J	1 1 1 0 0 0 0
К	0 0 1 1 0 0 1

Утверждается, что при кодировании символов с использованием данной кодовой таблицы появляется возможность однозначно декодировать значение символа по двоичной последовательности, в которой может быть допущена одна ошибка (один двоичный разряд изменил свое значение).

Были переданы три символа из указанного диапазона, закодированные с использованием кода приведенной кодовой таблицы. Известно, что при передаче кода каждого символа произошла ошибка в одном из двоичных разрядов. В результате было получено следующее сообщение:

1 0 1 0 0 1 1 1 0 1 1 0 0 1 1 1 1 1 0 0 1

Декодируйте полученное сообщение. В ответе укажите последовательность из трех символов в порядке их следования в сообщении без пробелов.

Вариант 4 Ответ: ДОБ

Код Хэмминга является примером кода с исправлением ошибок. Рассмотрим пример кодовой таблицы символов от 'А' до 'Р', сопоставляющей каждому символу двоичную последовательность, построенную на основе кода Хэмминга.

А	0 1 1 0 0 0 1
Б	1 0 1 1 0 0 0
В	0 0 0 0 0 0 0
Г	0 0 0 1 0 1 1
Д	0 1 0 0 1 1 1
Е	1 1 0 0 0 1 0
Ж	0 0 1 0 1 1 0
З	0 0 1 1 1 0 1
И	1 1 1 1 1 1 1
К	0 1 1 1 0 1 0
Л	1 1 1 0 1 0 0
М	1 0 0 1 1 1 0
Н	1 0 0 0 1 0 1
О	0 1 0 1 1 0 0
П	1 0 1 0 0 1 1
Р	1 1 0 1 0 0 1

Утверждается, что при кодировании символов с использованием данной кодовой таблицы появляется возможность однозначно декодировать значение символа по двоичной последовательности, в которой может быть допущена одна ошибка (один двоичный разряд изменил свое значение).

Были переданы три символа из указанного диапазона, закодированные с использованием кода приведенной кодовой таблицы. Известно, что при передаче кода каждого символа произошла ошибка в одном из двоичных разрядов. В результате было получено следующее сообщение:

0 1 0 0 1 1 0 0 1 0 0 1 0 0 1 0 1 1 1 0 0

Декодируйте полученное сообщение. В ответе укажите последовательность из трех символов в порядке их следования в сообщении без пробелов.

Вариант 5 Ответ: НМГ

Код Хэмминга является примером кода с исправлением ошибок. Рассмотрим пример кодовой таблицы символов от 'А' до 'Р', сопоставляющей каждому символу двоичную последовательность, построенную на основе кода Хэмминга.

А	0 1 1 0 0 0 1
Б	1 0 1 1 0 0 0
В	0 0 0 0 0 0 0
Г	0 0 0 1 0 1 1
Д	0 1 0 0 1 1 1
Е	1 1 0 0 0 1 0
Ж	0 0 1 0 1 1 0
З	0 0 1 1 1 0 1
И	1 1 1 1 1 1 1
К	0 1 1 1 0 1 0
Л	1 1 1 0 1 0 0

М	1 0 0 1 1 1 0
Н	1 0 0 0 1 0 1
О	0 1 0 1 1 0 0
П	1 0 1 0 0 1 1
Р	1 1 0 1 0 0 1

Утверждается, что при кодировании символов с использованием данной кодовой таблицы появляется возможность однозначно декодировать значение символа по двоичной последовательности, в которой может быть допущена одна ошибка (один двоичный разряд изменил свое значение).

Были переданы три символа из указанного диапазона, закодированные с использованием кода приведенной кодовой таблицы. Известно, что при передаче кода каждого символа произошла ошибка в одном из двоичных разрядов. В результате было получено следующее сообщение:

1 0 0 1 1 0 1 1 0 0 1 1 0 0 1 0 0 1 0 1 1

Декодируйте полученное сообщение. В ответе укажите последовательность из трех символов в порядке их следования в сообщении без пробелов.

Вариант 6 Ответ: РЕП

Код Хэмминга является примером кода с исправлением ошибок. Рассмотрим пример кодовой таблицы символов от 'А' до 'Р', сопоставляющей каждому символу двоичную последовательность, построенную на основе кода Хэмминга.

А	0 1 1 0 0 0 1
Б	1 0 1 1 0 0 0
В	0 0 0 0 0 0 0
Г	0 0 0 1 0 1 1
Д	0 1 0 0 1 1 1
Е	1 1 0 0 0 1 0
Ж	0 0 1 0 1 1 0
З	0 0 1 1 1 0 1
И	1 1 1 1 1 1 1
К	0 1 1 1 0 1 0
Л	1 1 1 0 1 0 0
М	1 0 0 1 1 1 0
Н	1 0 0 0 1 0 1
О	0 1 0 1 1 0 0
П	1 0 1 0 0 1 1
Р	1 1 0 1 0 0 1

Утверждается, что при кодировании символов с использованием данной кодовой таблицы появляется возможность однозначно декодировать значение символа по двоичной последовательности, в которой может быть допущена одна ошибка (один двоичный разряд изменил свое значение).

Были переданы три символа из указанного диапазона, закодированные с использованием кода приведенной кодовой таблицы. Известно, что при передаче кода каждого символа произошла ошибка в одном из двоичных разрядов. В результате было получено следующее сообщение:

1 1 0 0 0 0 1 1 1 0 0 0 0 0 1 1 1 0 0 1 1

Декодируйте полученное сообщение. В ответе укажите последовательность из трех символов в порядке их следования в сообщении без пробелов.

3. Основы логики (2 балла)

Вариант 1 Ответ: $A \text{ and not } B \text{ or } C \parallel \text{not } B \text{ and } A \text{ or } C \parallel C \text{ or } A \text{ and not } B \parallel C \text{ or } B \text{ and not } A$

Найдите логическую функцию, зависящую от трех логических переменных А, В и С, если известно, что:

1. Существует только три различных комбинации значений логических переменных, для которых значение функции будет «ложь».
2. Если значение логической переменной А принять за «ложь», то искомая функция станет эквивалентна логической функции $F(A,B,C) = C$.

3. Если значение логической переменной В принять за «ложь», то искомая функция станет эквивалентна логической функции $F(A,B,C) = A \text{ or } C$.
4. Если значение логической переменной С принять за «ложь», то искомая функция станет эквивалентна логической функции $F(A,B,C) = A \text{ and not } B$.

В ответе запишите формулу, которая может содержать логические переменные А, В и С и не более чем три логические операции.

*Комментарий по вводу ответа: операнды вводятся большими латинскими буквами; логические операции обозначаются, соответственно как **not**, **and** и **or**.*

Пример записи ответа: A or B and C

Вариант 2 Ответ: not A and C or B || C and not A or B || B or not A and C || B or C and not A

Найдите логическую функцию, зависящую от трех логических переменных А, В и С, если известно, что:

1. Существует только три различных комбинации значений логических переменных, для которых значение функции будет «ложь».
2. Если значение логической переменной А принять за «ложь», то искомая функция станет эквивалентна логической функции $F(A,B,C) = B \text{ or } C$.
3. Если значение логической переменной В принять за «ложь», то искомая функция станет эквивалентна логической функции $F(A,B,C) = \text{not } A \text{ and } C$.
4. Если значение логической переменной С принять за «ложь», то искомая функция станет эквивалентна логической функции $F(A,B,C) = B$.

В ответе запишите формулу, которая может содержать логические переменные А, В и С и не более чем три логические операции.

*Комментарий по вводу ответа: операнды вводятся большими латинскими буквами; логические операции обозначаются, соответственно как **not**, **and** и **or**.*

Пример записи ответа: A or B and C

Вариант 3 Ответ: not B or A and C || not B or C and A || A and C or not B || C and A or not B

Найдите логическую функцию, зависящую от трех логических переменных А, В и С, если известно, что:

1. Существует только три различных комбинации значений логических переменных, для которых значение функции будет «ложь».
2. Если значение логической переменной А принять за «ложь», то искомая функция станет эквивалентна логической функции $F(A,B,C) = \text{not } B$.
3. Если значение логической переменной В принять за «истина», то искомая функция станет эквивалентна логической функции $F(A,B,C) = A \text{ and } C$.
4. Если значение логической переменной С принять за «истина», то искомая функция станет эквивалентна логической функции $F(A,B,C) = A \text{ or not } B$.

В ответе запишите формулу, которая может содержать логические переменные А, В и С и не более чем три логические операции.

*Комментарий по вводу ответа: операнды вводятся большими латинскими буквами; логические операции обозначаются, соответственно как **not**, **and** и **or**.*

Пример записи ответа: A or B and C

Вариант 4 Ответ: not A and B or C || B and not A or C || C or not A and B || C or B and not A

Найдите логическую функцию, зависящую от трех логических переменных А, В и С, если известно, что:

1. Существует только три различных комбинации значений логических переменных, для которых значение функции будет «ложь».

2. Если значение логической переменной А принять за «ложь», то искомая функция будет принимать значение «ложь» только для одной комбинации значений логических переменных.
3. Если значение логической переменной В принять за «ложь», то искомая функция будет принимать значение «ложь» только для двух различных комбинаций значений логических переменных.
4. Если значение логической переменной С принять за «истина», то искомая функция будет принимать значение «истина» для всех комбинаций значений логических переменных.

В ответе запишите формулу, которая может содержать логические переменные А, В и С и не более чем три логические операции.

*Комментарий по вводу ответа: операнды вводятся большими латинскими буквами; логические операции обозначаются, соответственно как **not**, **and** и **or**.*

Пример записи ответа: A or B and C

Вариант 5 Ответ: A or not B and C || A or C and not B || not B and C or A || C and not B or A

Найдите логическую функцию, зависящую от трех логических переменных А, В и С, если известно, что:

1. Существует только три различных комбинации значений логических переменных, для которых значение функции будет «ложь».
2. Если значение логической переменной А принять за «истина», то искомая функция будет принимать значение «истина» для всех комбинаций значений логических переменных.
3. Если значение логической переменной В принять за «ложь», то искомая функция будет принимать значение «ложь» только для одной комбинации значений логических переменных.
4. Если значение логической переменной С принять за «ложь», то искомая функция будет принимать значение «ложь» только для двух различных комбинаций значений логических переменных.

В ответе запишите формулу, которая может содержать логические переменные А, В и С и не более чем три логические операции.

*Комментарий по вводу ответа: операнды вводятся большими латинскими буквами; логические операции обозначаются, соответственно как **not**, **and** и **or**.*

Пример записи ответа: A or B and C

Вариант 6 Ответ: B or A and not C || B or not C and A || A and not C or B || not C and A or B

Найдите логическую функцию, зависящую от трех логических переменных А, В и С, если известно, что:

1. Существует только три различных комбинации значений логических переменных, для которых значение функции будет «ложь».
2. Если значение логической переменной В принять за «истина», то искомая функция будет принимать значение «истина» для всех комбинаций значений логических переменных.
3. Если значение логической переменной С принять за «ложь», то искомая функция будет принимать значение «ложь» только для одной комбинации значений логических переменных.
4. Если значение логической переменной А принять за «ложь», то искомая функция будет принимать значение «ложь» только для двух различных комбинаций значений логических переменных.

В ответе запишите формулу, которая может содержать логические переменные А, В и С и не более чем три логические операции.

*Комментарий по вводу ответа: операнды вводятся большими латинскими буквами; логические операции обозначаются, соответственно как **not**, **and** и **or**.*

Пример записи ответа: A or B and C

4. Алгоритмизация и программирование (1 балл)

Вариант 1 Ответ: 2 3 3

Дана исходная последовательность цифр: 1234

Задан алгоритм преобразования последовательности, на каждом шаге которого выполняются следующие операции:

1. В конец последовательности, имеющейся перед выполнением шага, дописывается ее копия, но развернутая зеркально (цифры записываются в обратном порядке).
2. В конце получившейся последовательности удаляется количество цифр, равное номеру шага выполнения алгоритма.

Ниже приведены результаты выполнения первых двух шагов алгоритма:

1: 1234432

2: 123443223443

Определите, какие цифры будут на 101-ой, 301-ой и 501-ой позиции от начала последовательности, которая получилась после выполнения 8-ого шага алгоритма.

В ответе укажите через пробел три цифры: сначала цифру, которая стоит на 101-ой позиции, затем цифру, которая стоит на 301-ой позиции и затем цифру, которая стоит на 501-ой позиции.

Вариант 2 Ответ: 4 2 4

Дана исходная последовательность цифр: 12345

Задан алгоритм преобразования последовательности, на каждом шаге которого выполняются следующие операции:

1. В конец последовательности, имеющейся перед выполнением шага, дописывается ее копия, но развернутая зеркально (цифры записываются в обратном порядке).
2. В конце получившейся последовательности удаляется количество цифр, равное номеру шага выполнения алгоритма.

Ниже приведены результаты выполнения первых двух шагов алгоритма:

1: 12345432

2: 12345432234543

Определите, какие цифры будут на 102-ой, 302-ой и 502-ой позиции от начала последовательности, которая получилась после выполнения 8-ого шага алгоритма.

В ответе укажите через пробел три цифры: сначала цифру, которая стоит на 102-ой позиции, затем цифру, которая стоит на 302-ой позиции и затем цифру, которая стоит на 502-ой позиции.

Вариант 3 Ответ: 4 2 5

Дана исходная последовательность цифр: 123456

Задан алгоритм преобразования последовательности, на каждом шаге которого выполняются следующие операции:

3. В конец последовательности, имеющейся перед выполнением шага, дописывается ее копия, но развернутая зеркально (цифры записываются в обратном порядке).
4. В конце получившейся последовательности удаляется количество цифр, равное номеру шага выполнения алгоритма.

Ниже приведены результаты выполнения первых двух шагов алгоритма:

1: 12345665432

2: 12345665432234566543

Определите, какие цифры будут на 103-ой, 303-ой и 503-ой позиции от начала последовательности, которая получилась после выполнения 7-ого шага алгоритма.

В ответе укажите через пробел три цифры: сначала цифру, которая стоит на 103-ой позиции, затем цифру, которая стоит на 303-ой позиции и затем цифру, которая стоит на 503-ой позиции.

Вариант 4 Ответ: 4 2 1

Дана исходная последовательность цифр: 11223344

Задан алгоритм преобразования последовательности, на каждом шаге которого выполняются следующие операции:

1. В конец последовательности, имеющейся перед выполнением шага, дописывается ее копия, но развернутая зеркально (цифры записываются в обратном порядке).

2. В начале получившейся последовательности удаляется количество цифр, равное номеру шага выполнения алгоритма.
3. В конце получившейся последовательности удаляется количество цифр, равное номеру шага выполнения алгоритма.

Ниже приведены результаты выполнения первых двух шагов алгоритма:

1: 12233444433221

2: 233444433221122334444332

Определите, какие цифры будут на 111-ой, 333-ой и 444-ой позиции от начала последовательности, которая получилась после выполнения 7-ого шага алгоритма.

В ответе укажите через пробел три цифры: сначала цифру, которая стоит на 111-ой позиции, затем цифру, которая стоит на 333-ой позиции и затем цифру, которая стоит на 444-ой позиции.

Вариант 5 Ответ: 5 4 2

Дана исходная последовательность цифр: 1122334455

Задан алгоритм преобразования последовательности, на каждом шаге которого выполняются следующие операции:

1. В конец последовательности, имеющейся перед выполнением шага, дописывается ее копия, но развернутая зеркально (цифры записываются в обратном порядке).
2. В начале получившейся последовательности удаляется количество цифр, равное номеру шага выполнения алгоритма.
3. В конце получившейся последовательности удаляется количество цифр, равное номеру шага выполнения алгоритма.

Ниже приведены результаты выполнения первых двух шагов алгоритма:

1: 12233445554433221

2: 233445554433221122334455544332

Определите, какие цифры будут на 222-ой, 444-ой и 555-ой позиции от начала последовательности, которая получилась после выполнения 7-ого шага алгоритма.

В ответе укажите через пробел три цифры: сначала цифру, которая стоит на 222-ой позиции, затем цифру, которая стоит на 444-ой позиции и затем цифру, которая стоит на 555-ой позиции.

Вариант 6 Ответ: 4 6 5

Дана исходная последовательность цифр: 112233445566

Задан алгоритм преобразования последовательности, на каждом шаге которого выполняются следующие операции:

1. В конец последовательности, имеющейся перед выполнением шага, дописывается ее копия, но развернутая зеркально (цифры записываются в обратном порядке).
2. В начале получившейся последовательности удаляется количество цифр, равное номеру шага выполнения алгоритма.
3. В конце получившейся последовательности удаляется количество цифр, равное номеру шага выполнения алгоритма.

Ниже приведены результаты выполнения первых двух шагов алгоритма:

1: 122334455666554433221

2: 23344556665544332211223344556665544332

Определите, какие цифры будут на 100-ой, 300-ой и 500-ой позиции от начала последовательности, которая получилась после выполнения 6-ого шага алгоритма.

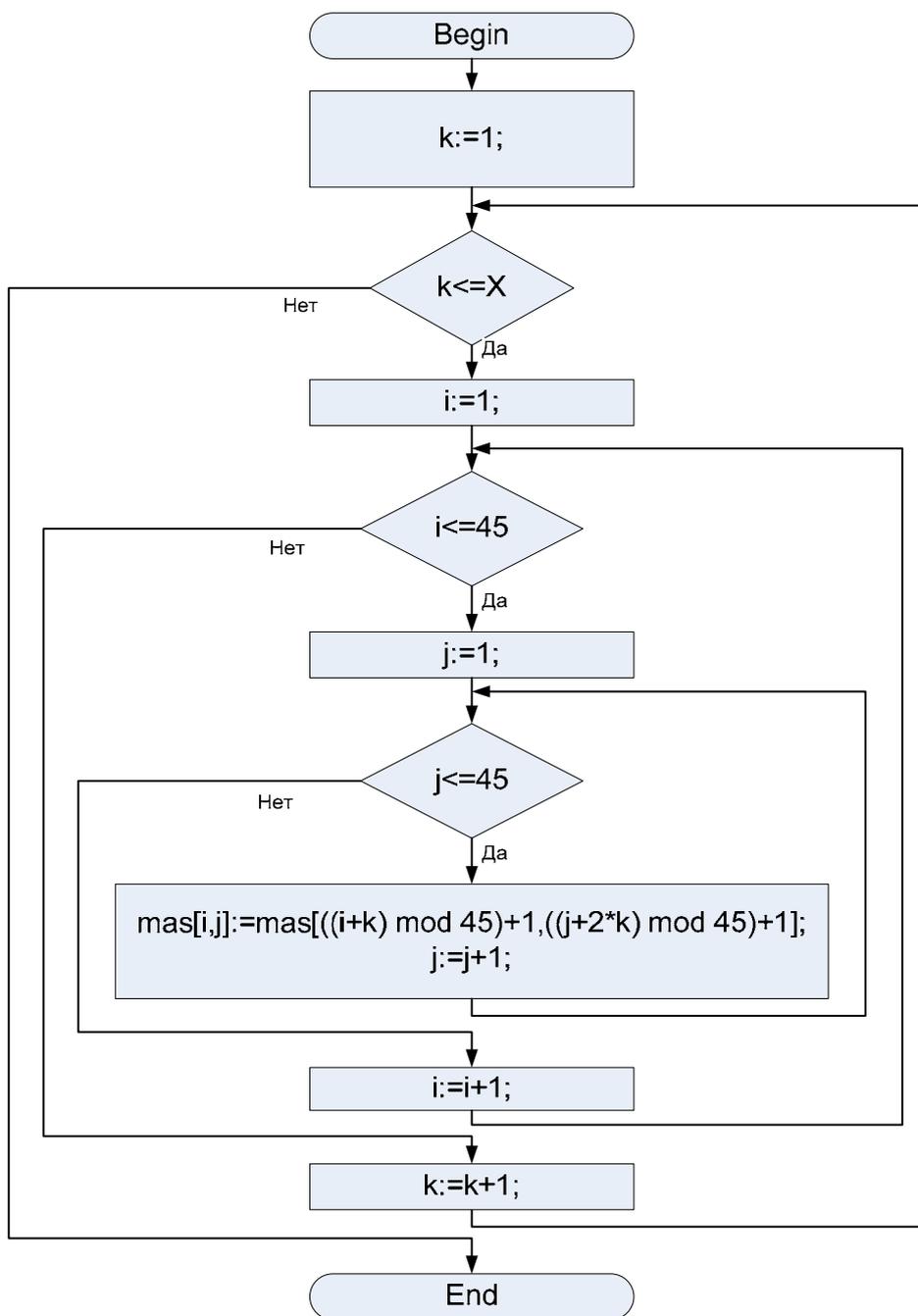
В ответе укажите через пробел три цифры: сначала цифру, которая стоит на 100-ой позиции, затем цифру, которая стоит на 300-ой позиции и затем цифру, которая стоит на 500-ой позиции.

5. Алгоритмизация и программирование (2 балла)

Вариант 1 Ответ: 87

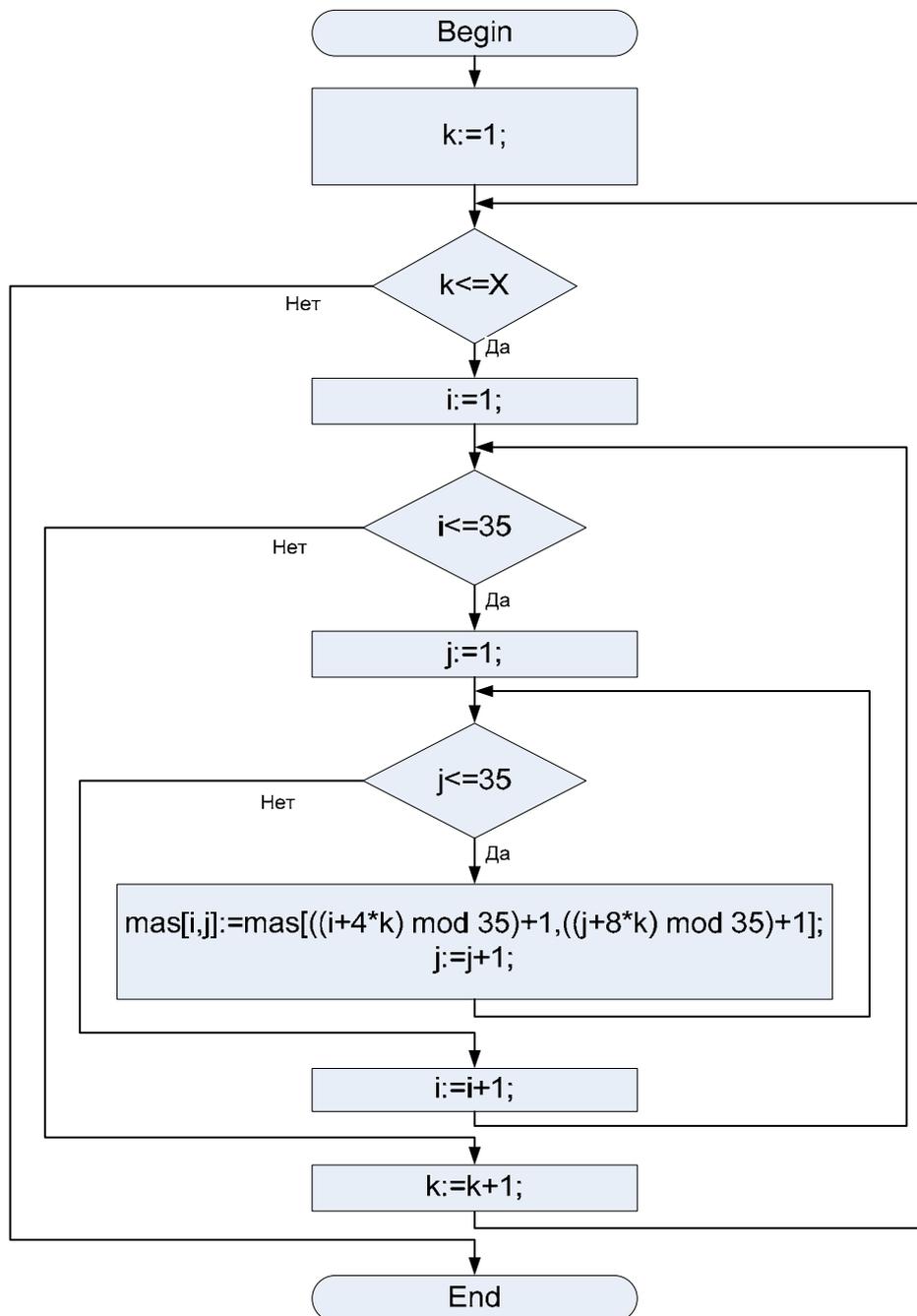
Дана блок-схема алгоритма обработки двумерного массива **mas**, размером 45 на 45 элементов. Перед началом выполнения алгоритма массив заполнен неповторяющимися целочисленными элементами. Какое минимальное значение целочисленной переменной **X** должно быть на входе алгоритма, чтобы после его выполнения все элементы массива имели одинаковые значения? При обращении к элементам

массива первый индекс обозначает номер строки, а второй индекс – номер столбца. Нумерация элементов массива начинается с [1,1]. В ответ напишите число.



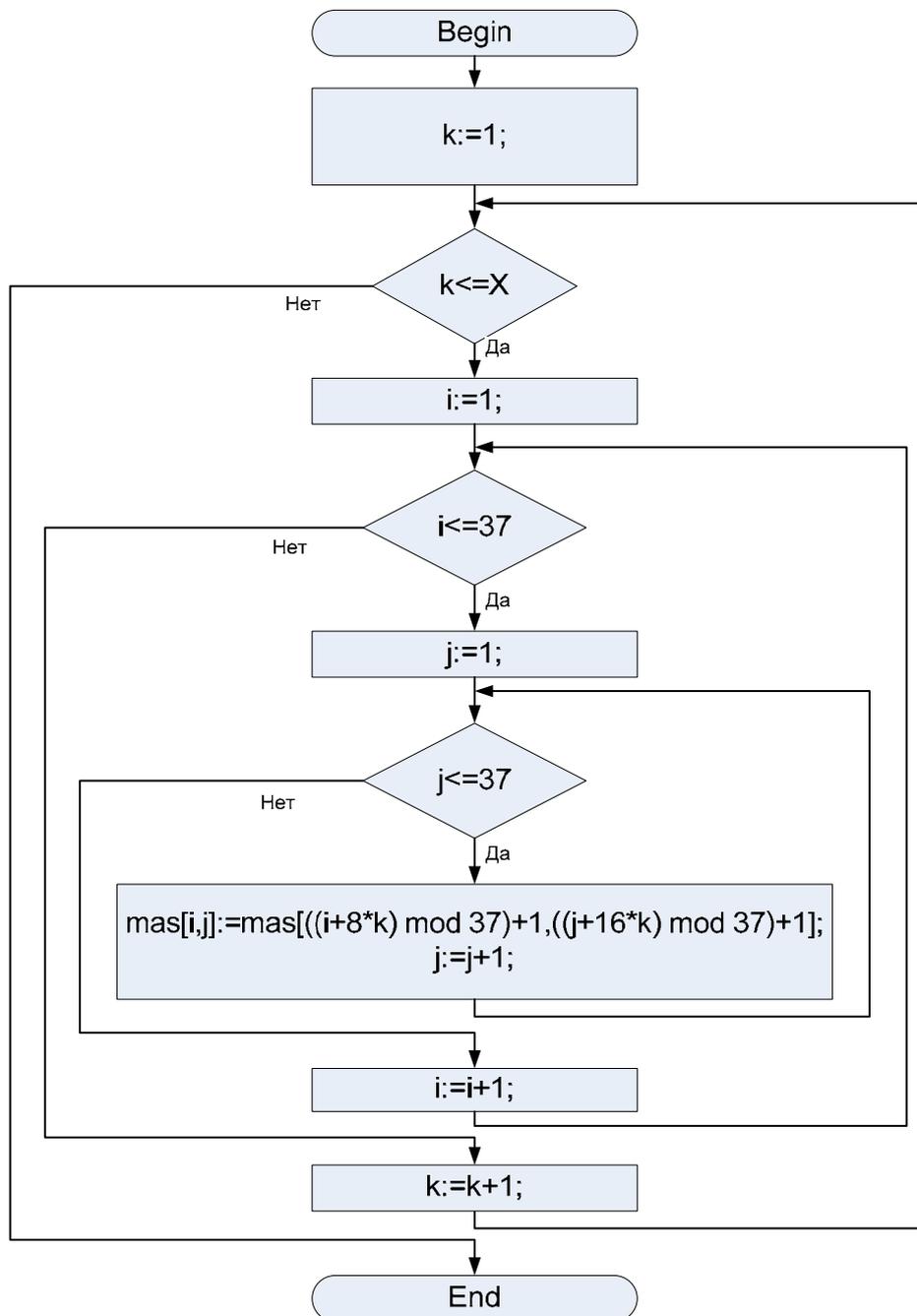
Вариант 2 Ответ: 52

Дана блок-схема алгоритма обработки двумерного массива **mas**, размером 35 на 35 элементов. Перед началом выполнения алгоритма массив заполнен неповторяющимися целочисленными элементами. Какое минимальное значение целочисленной переменной **X** должно быть на входе алгоритма, чтобы после его выполнения все элементы массива имели одинаковые значения? При обращении к элементам массива первый индекс обозначает номер строки, а второй индекс – номер столбца. Нумерация элементов массива начинается с [1,1]. В ответ напишите число.



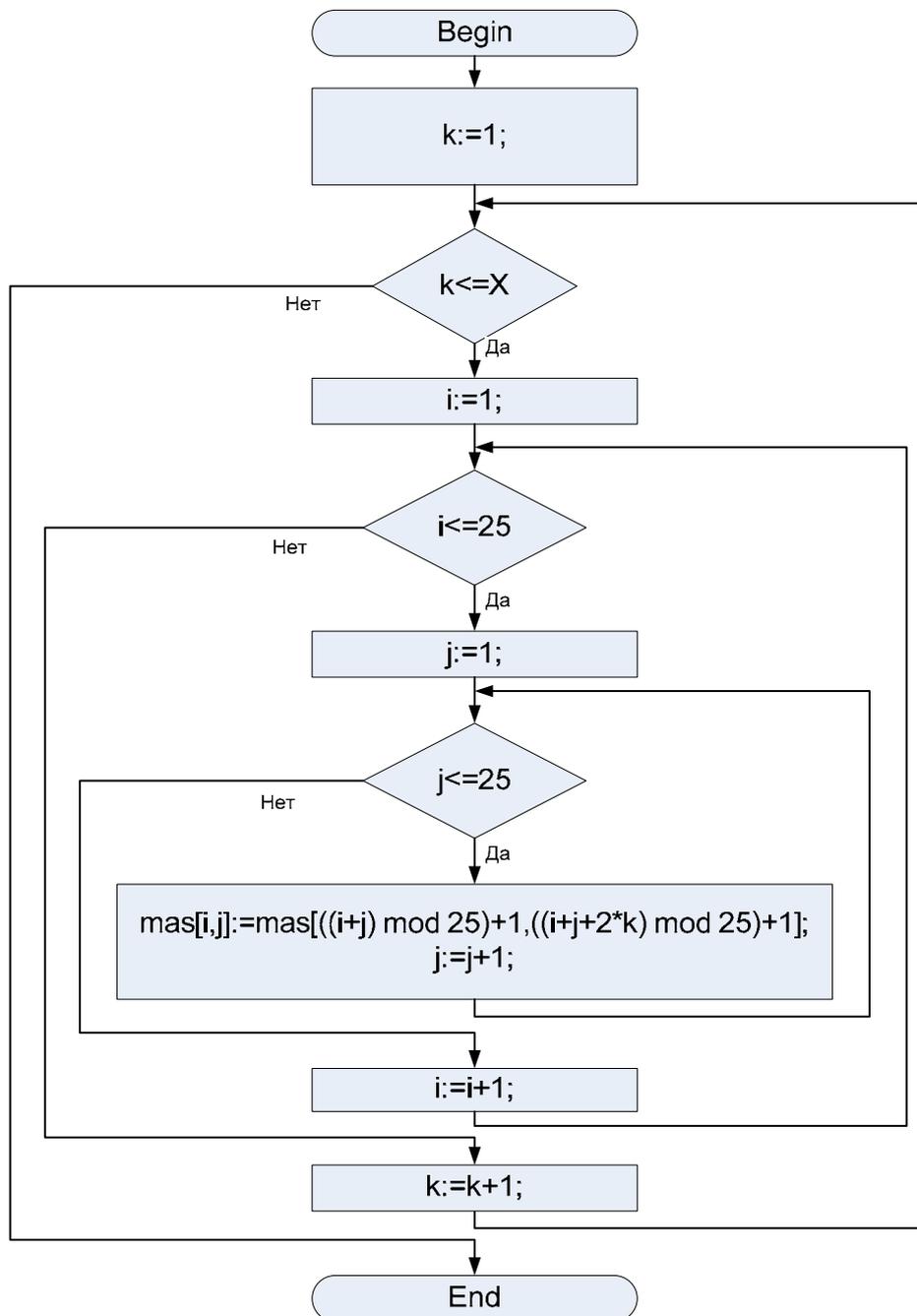
Вариант 3 Ответ: 46

Дана блок-схема алгоритма обработки двумерного массива **mas**, размером 37 на 37 элементов. Перед началом выполнения алгоритма массив заполнен неповторяющимися целочисленными элементами. Какое минимальное значение целочисленной переменной **X** должно быть на входе алгоритма, чтобы после его выполнения все элементы массива имели одинаковые значения? При обращении к элементам массива первый индекс обозначает номер строки, а второй индекс – номер столбца. Нумерация элементов массива начинается с [1,1]. В ответ напишите число.



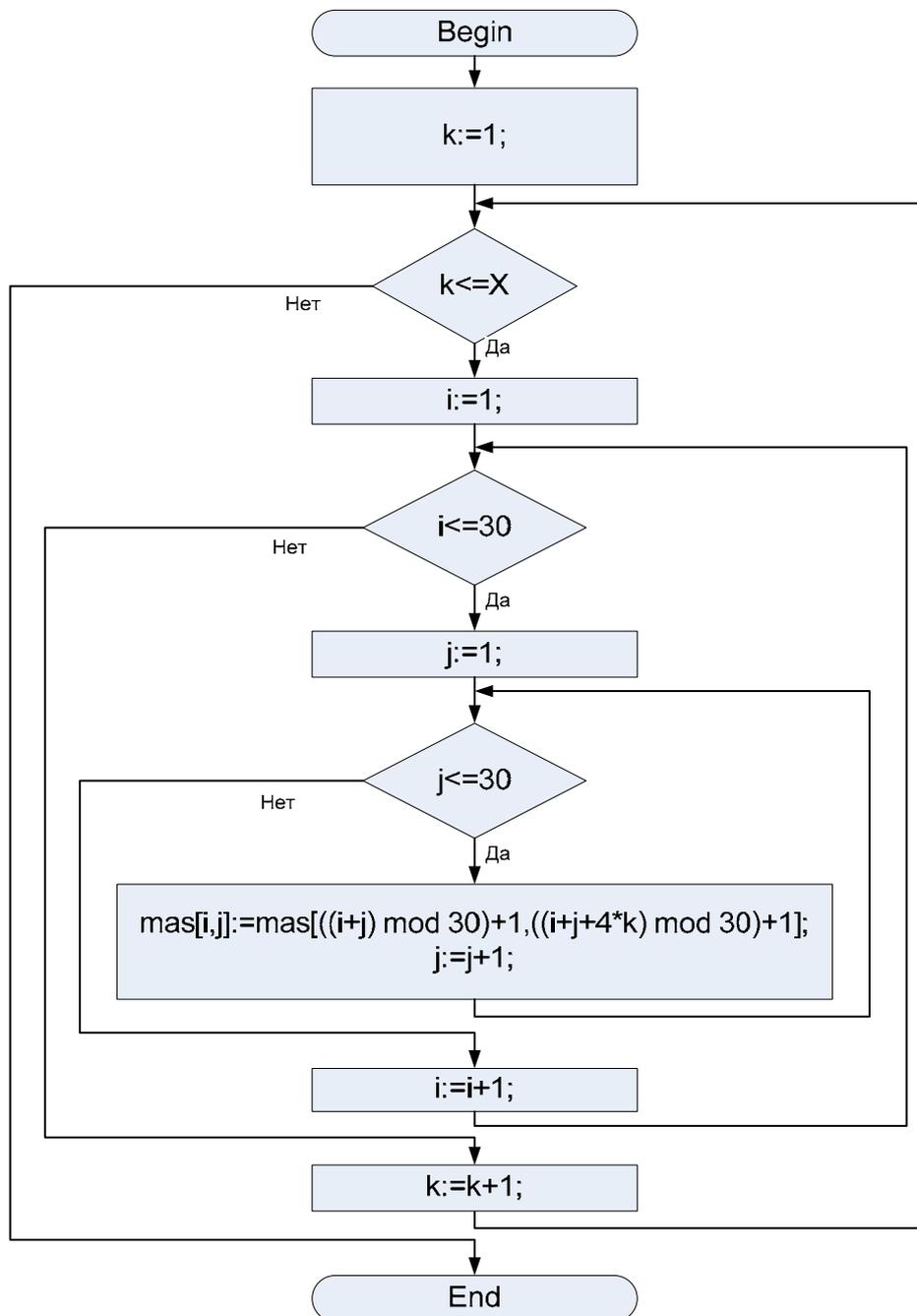
Вариант 4 Ответ: 16

Дана блок-схема алгоритма обработки двумерного массива **mas**, размером 25 на 25 элементов. Перед началом выполнения алгоритма массив заполнен неповторяющимися целочисленными элементами. Какое минимальное значение целочисленной переменной **X** должно быть на входе алгоритма, чтобы после его выполнения все элементы массива имели одинаковые значения? При обращении к элементам массива первый индекс обозначает номер строки, а второй индекс – номер столбца. Нумерация элементов массива начинается с [1,1]. В ответ напишите число.



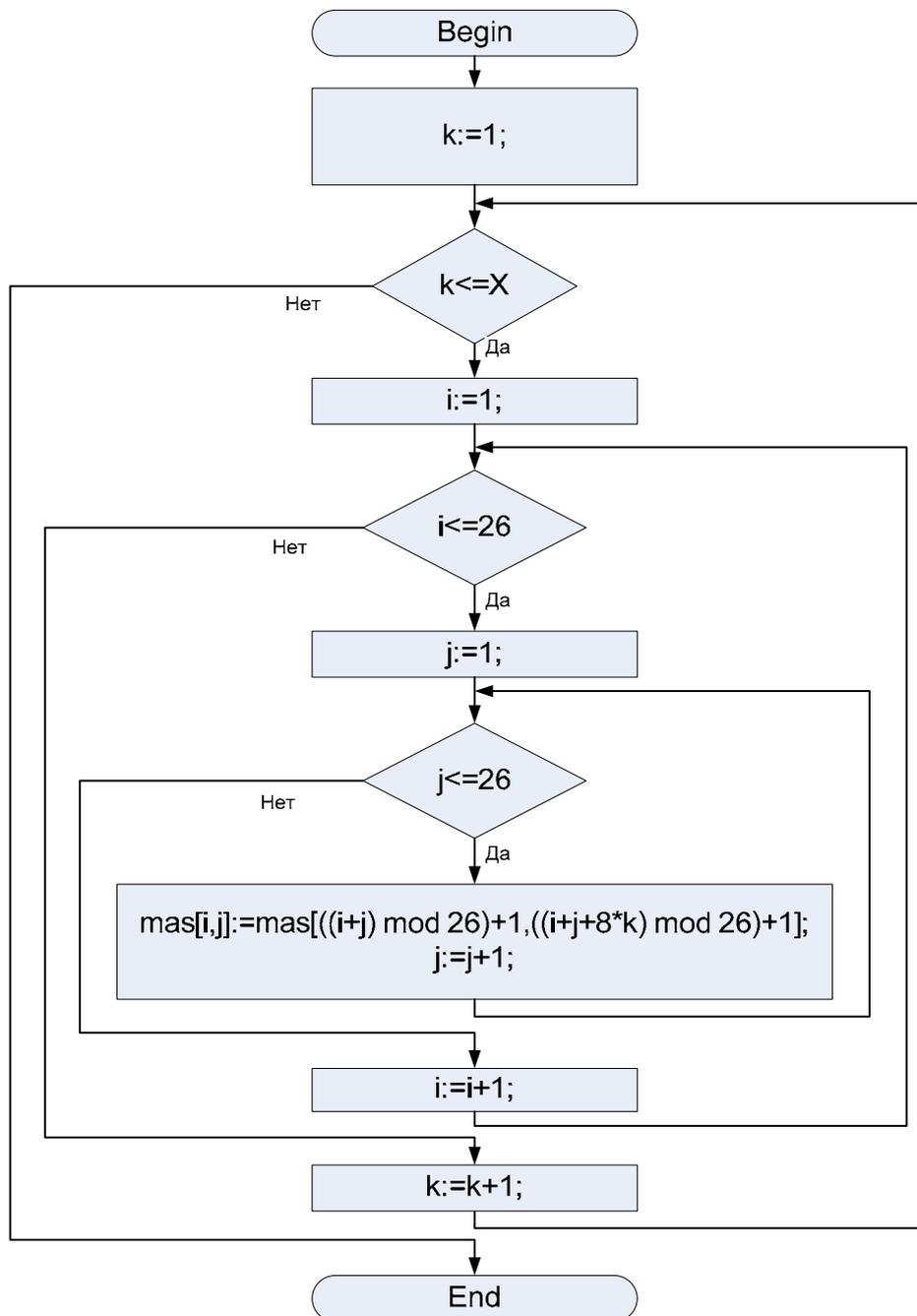
Вариант 5 Ответ: 18

Дана блок-схема алгоритма обработки двумерного массива **mas**, размером 30 на 30 элементов. Перед началом выполнения алгоритма массив заполнен неповторяющимися целочисленными элементами. Какое минимальное значение целочисленной переменной **X** должно быть на входе алгоритма, чтобы после его выполнения все элементы массива имели одинаковые значения? При обращении к элементам массива первый индекс обозначает номер строки, а второй индекс – номер столбца. Нумерация элементов массива начинается с [1,1]. В ответ напишите число.



Вариант 6 Ответ: 34

Дана блок-схема алгоритма обработки двумерного массива **mas**, размером 26 на 26 элементов. Перед началом выполнения алгоритма массив заполнен неповторяющимися целочисленными элементами. Какое минимальное значение целочисленной переменной **X** должно быть на входе алгоритма, чтобы после его выполнения все элементы массива имели одинаковые значения? При обращении к элементам массива первый индекс обозначает номер строки, а второй индекс – номер столбца. Нумерация элементов массива начинается с [1,1]. В ответ напишите число.



6. Алгоритмизация и программирование (1 балл)

Вариант 1 Ответ: 77

Дан фрагмент программы:

Бейсик	Паскаль	Алгоритмический
<pre> R = 0 WHILE X <> 4 R=R*10+(X MOD 2) X=X \ 2 WEND </pre>	<pre> R:=0; while X<>4 do begin R:=R*10+(X mod 2); X:=X div 2; end; </pre>	<pre> R:=0; пока X<>4 <u>нц</u> R:=R * 10 + ост_дел(X,2) X:= цел_дел(X,2) <u>кц</u> </pre>

Операции MOD, mod и функция ост_дел вычисляют остаток от деления первого аргумента на второй.

Операции \, div и функция цел_дел осуществляют целочисленное деление.

Какое минимальное значение целочисленной переменной **X** должно было быть перед началом выполнения этого фрагмента, если после его выполнения получилось значение **R=1011**?

В ответе укажите целое число.

Вариант 2 Ответ: 139

Дан фрагмент программы:

Бейсик	Паскаль	Алгоритмический
<pre>R = 0 WHILE X<>8 R=R*10+(X MOD 2) X=X \ 2 WEND</pre>	<pre>R:=0; while X<>8 do begin R:=R*10+(X mod 2); X:=X div 2; end;</pre>	<pre>R:=0; пока X<>8 <u>нц</u> R:=R * 10 + ост_дел(X, 2) X:= цел_дел(X, 2) <u>кц</u></pre>

Операции MOD, mod и функция ост_дел вычисляют остаток от деления первого аргумента на второй.

Операции \, div и функция цел_дел осуществляют целочисленное деление.

Какое минимальное значение целочисленной переменной **X** должно было быть перед началом выполнения этого фрагмента, если после его выполнения получилось значение **R=1101**?

В ответе укажите целое число.

Вариант 3 Ответ: 265

Дан фрагмент программы:

Бейсик	Паскаль	Алгоритмический
<pre>R = 0 WHILE X<>16 R=R*10+(X MOD 2) X=X \ 2 WEND</pre>	<pre>R:=0; while X<>16 do begin R:=R*10+(X mod 2); X:=X div 2; end;</pre>	<pre>R:=0; пока X<>16 <u>нц</u> R:=R * 10 + ост_дел(X, 2) X:= цел_дел(X, 2) <u>кц</u></pre>

Операции MOD, mod и функция ост_дел вычисляют остаток от деления первого аргумента на второй.

Операции \, div и функция цел_дел осуществляют целочисленное деление.

Какое минимальное значение целочисленной переменной **X** должно было быть перед началом выполнения этого фрагмента, если после его выполнения получилось значение **R=1001**?

В ответе укажите целое число.

Вариант 4 Ответ: 4196

Дан фрагмент программы:

Бейсик	Паскаль	Алгоритмический
<pre>R = 0 WHILE X<>4 R=R*10+(X MOD 4)+1 X=X \ 4 WEND</pre>	<pre>R:=0; while X<>4 do begin R:=R*10+(X mod 4)+1; X:=X div 4; end;</pre>	<pre>R:=0; пока X<>4 <u>нц</u> R:=R * 10 + ост_дел(X, 4)+1 X:= цел_дел(X, 4) <u>кц</u></pre>

Операции MOD, mod и функция ост_дел вычисляют остаток от деления первого аргумента на второй.

Операции \, div и функция цел_дел осуществляют целочисленное деление.

Какое минимальное значение целочисленной переменной **X** должно было быть перед началом выполнения этого фрагмента, если после его выполнения получилось значение **R=12321**?

В ответе укажите целое число.

Вариант 5 Ответ: 8774

Дан фрагмент программы:

Бейсик	Паскаль	Алгоритмический
<pre>R = 0 WHILE X<>8 R=R*10+(X MOD 4)+1 X=X \ 4 WEND</pre>	<pre>R:=0; while X<>8 do begin R:=R*10+(X mod 4)+1; X:=X div 4; end;</pre>	<pre>R:=0; пока X<>8 <u>нц</u> R:=R * 10 + ост_дел(X, 4)+1 X:= цел_дел(X, 4) <u>кц</u></pre>

Операции MOD, mod и функция ост_дел вычисляют остаток от деления первого аргумента на второй.

Операции \, div и функция цел_дел осуществляют целочисленное деление.

Какое минимальное значение целочисленной переменной **X** должно было быть перед началом выполнения этого фрагмента, если после его выполнения получилось значение **R=32123**?

В ответе укажите целое число.

Вариант 6 Ответ: 4246

Дан фрагмент программы:

Бейсик	Паскаль	Алгоритмический
<pre>R = 0 WHILE X<>16 R=R*10+(X MOD 4) +1 X=X \ 4 WEND</pre>	<pre>R:=0; while X<>16 do begin R:=R*10+(X mod 4) +1; X:=X div 4; end;</pre>	<pre>R:=0; пока X<>16 <u>нц</u> R:=R * 10 + ост_дел(X, 4) +1 X:= цел_дел(X, 4) <u>кц</u></pre>

Операции MOD, mod и функция ост_дел вычисляют остаток от деления первого аргумента на второй.

Операции \, div и функция цел_дел осуществляют целочисленное деление.

Какое минимальное значение целочисленной переменной **X** должно было быть перед началом выполнения этого фрагмента, если после его выполнения получилось значение **R=3223**?

В ответе укажите целое число.

7. Кодирование графической и звуковой информации (2 балла)

Вариант 1 Ответ: 25

Для хранения отсканированных слайдов отведено **X** МБайт памяти (**X** – целое число). Если сканирование слайдов произвести с использованием палитры из 2^{32} цветов, то отведенной памяти хватит на хранение семи изображений и останутся свободными 2,25 МБайт. Если сканирование слайдов произвести с использованием палитры из 2^{24} цветов, то отведенной памяти хватит на хранение десяти изображений и отведенный объем памяти будет использован полностью. Сканированные изображения содержат одинаковое количество пикселей. Известно, что каждый файл с изображением кроме несжатой графической информации также содержит 256 КБайт служебной информации. Сколько памяти отведено для хранения сканированных изображений? В ответе укажите целое число МБайт.

Примечание: 1 МБайт = 1024 КБайт. 1 КБайт = 1024 Байт.

Вариант 2 Ответ: 65

Для хранения отсканированных слайдов отведено **X** МБайт памяти (**X** – целое число). Если сканирование слайдов произвести с использованием палитры из 2^{32} цветов, то отведенной памяти хватит на хранение семи изображений и останутся свободными 5,5 МБайт. Если сканирование слайдов произвести с использованием палитры из 2^{24} цветов, то отведенной памяти хватит на хранение десяти изображений и отведенный объем памяти будет использован полностью. Сканированные изображения содержат одинаковое количество пикселей. Известно, что каждый файл с изображением кроме несжатой графической информации также содержит 512 КБайт служебной информации. Сколько памяти отведено для хранения сканированных изображений? В ответе укажите целое число МБайт.

Примечание: 1 МБайт = 1024 КБайт. 1 КБайт = 1024 Байт.

Вариант 3 Ответ: 95

Для хранения отсканированных слайдов отведено **X** МБайт памяти (**X** – целое число). Если сканирование слайдов произвести с использованием палитры из 2^{32} цветов, то отведенной памяти хватит на хранение семи изображений и останутся свободными 7,5 МБайт. Если сканирование слайдов произвести с использованием палитры из 2^{24} цветов, то отведенной памяти хватит на хранение десяти изображений и отведенный объем памяти будет использован полностью. Сканированные изображения содержат одинаковое количество пикселей. Известно, что каждый файл с изображением кроме несжатой графической информации также содержит 512 КБайт служебной информации. Сколько памяти отведено для хранения сканированных изображений? В ответе укажите целое число МБайт.

Примечание: 1 МБайт = 1024 КБайт. 1 КБайт = 1024 Байт.

Вариант 4 Ответ: 1048

Для хранения оцифрованных фрагментов звукозаписей отведено X МБайт памяти (X – целое число). Если оцифровку проводить с частотой дискретизации 32000 Hz, то отведенной памяти хватит на хранение шестидесяти пяти фрагментов и останется свободным 8 МБайт. Если оцифровку проводить с частотой дискретизации 16000 Hz, то отведенной памяти хватит на хранение ста двадцати восьми фрагментов и отведенный объем памяти будет использован полностью. Фрагменты звукозаписей имеют одинаковую продолжительность звучания. Оцифровка проводится с использованием 65536 уровней квантования в стереофоническом режиме. При кодировании звука сжатия не производится. Известно, что каждый файл с оцифрованным фрагментом дополнительно содержит 384 КБайт служебной информации. Сколько памяти отведено для хранения оцифрованных фрагментов звукозаписей? В ответе укажите целое число МБайт.

Примечание: 1 МБайт = 1024 КБайт. 1 КБайт = 1024 Байт.

Вариант 5 Ответ: 2064

Для хранения оцифрованных фрагментов звукозаписей отведено X МБайт памяти (X – целое число). Если оцифровку проводить с частотой дискретизации 32000 Hz, то отведенной памяти хватит на хранение ста тридцати фрагментов и останется свободным 0,25 МБайт. Если оцифровку проводить с частотой дискретизации 16000 Hz, то отведенной памяти хватит на хранение двухсот пятидесяти шести фрагментов и отведенный объем памяти будет использован полностью. Фрагменты звукозаписей имеют одинаковую продолжительность звучания. Оцифровка проводится с использованием 65536 уровней квантования в стереофоническом режиме. При кодировании звука сжатия не производится. Известно, что каждый файл с оцифрованным фрагментом дополнительно содержит 256 КБайт служебной информации. Сколько памяти отведено для хранения оцифрованных фрагментов звукозаписей? В ответе укажите целое число МБайт.

Примечание: 1 МБайт = 1024 КБайт. 1 КБайт = 1024 Байт.

Вариант 6 Ответ: 1032

Для хранения оцифрованных фрагментов звукозаписей отведено X МБайт памяти (X – целое число). Если оцифровку проводить с частотой дискретизации 32000 Hz, то отведенной памяти хватит на хранение шестидесяти пяти фрагментов и останется свободным 0,125 МБайт. Если оцифровку проводить с частотой дискретизации 16000 Hz, то отведенной памяти хватит на хранение ста двадцати восьми фрагментов и отведенный объем памяти будет использован полностью. Фрагменты звукозаписей имеют одинаковую продолжительность звучания. Оцифровка проводится с использованием 65536 уровней квантования в стереофоническом режиме. При кодировании звука сжатия не производится. Известно, что каждый файл с оцифрованным фрагментом дополнительно содержит 256 КБайт служебной информации. Сколько памяти отведено для хранения оцифрованных фрагментов звукозаписей? В ответе укажите целое число МБайт.

Примечание: 1 МБайт = 1024 КБайт. 1 КБайт = 1024 Байт.

8. Телекоммуникационные технологии (3 балла)

Вариант 1 Ответ: 256

Студенты Иван и Петр сконструировали прототип компьютерной сети и реализовали стек протоколов, основывающийся на принципах стандарта IEEE 802.5.

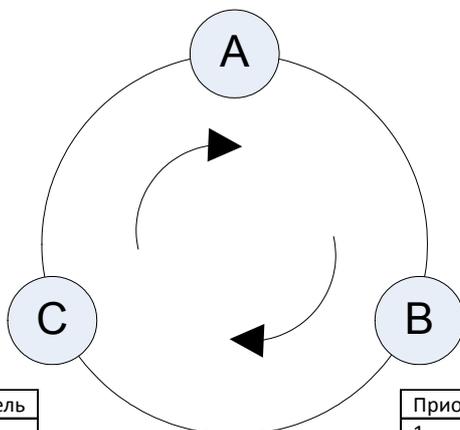
Сеть имеет топологию кольцо (все узлы объединены последовательно в кольцо, а данные передаются по кругу в одном направлении, при этом каждый узел ретранслирует все проходящие кадры) и использует маркерный метод доступа. В сети циркулирует маркер (специальный кадр стандартного размера). Узел, получивший маркер, может начать передачу данных, для чего изымает маркер из сети и выпускает кадр данных, который передается последующими в кольце узлами, пока не попадет на узел-получатель, тот выставит в кадре признак приема и передаст кадр по кольцу далее, до узла отправителя. Тот в свою очередь изымет кадр из сети и либо освободит маркер, давая следующему по кольцу узлу начать передачу, либо продолжит удерживать маркер и начнет передачу следующего кадра из своей очереди, согласно представленным ниже правилам:

1. Узел будет удерживать маркер до тех пор, пока не опустеет его очередь или пока не истечет время удержания маркера данным узлом. Время удержания маркера узлом определяется как разность номинального времени обхода кольца (ВНО), задаваемого в виде константы в момент инициализации кольца, и времени реального отсутствия маркера на данном узле (ВРО). Время реального отсутствия маркера на узле определяется как интервал между моментом, когда маркер покинул данный узел в прошлый раз и моментом, когда узел принял маркер в следующий раз.

2. Имеется два типа трафика – с высшим приоритетом (1) и с обычным приоритетом (0). Признаком приоритета данные помечаются в очереди сетевого интерфейса, но приоритет влияет только на порядок передачи данных через сетевой интерфейс, порядок данных в очереди своего узла от приоритета не зависит и остается неизменным.
3. Когда узел получает маркер первый раз (от начала работы сети) – он может передать только один кадр (независимо от его приоритета), после чего обязан выпустить маркер дальше.
4. В общем случае, когда узел получает маркер, на узле вычисляется время удержания маркера. Если получившееся время (ВНО-ВРО) меньше или равно нулю, то узел передает только один кадр с высшим приоритетом, если такой стоит следующим в его очереди или пропускает маркер дальше. Если время удержания маркера на данном узле получается положительным, то узел начинает передачу данных из своей очереди по порядку и независимо от их приоритета.
5. Узел может продолжать передачу данных в течение времени удержания маркера, вычисленного в момент прихода маркера на узел, причем, если этот период истек до прихода очередного кадра с подтверждением приема, узел дожидается этого события, а потом выпускает маркер вне зависимости от приоритета передаваемых данных.

Схема сети и очереди сетевых интерфейсов представлены на рисунке (началу очередей соответствуют первые строки таблиц):

Приоритет	Объем данных	Получатель
0	1 КБайт	В
1	16 КБайт	В
0	9 КБайт	С



Приоритет	Объем данных	Получатель
0	1 КБайт	А
0	8 КБайт	В
1	2 КБайт	А

Приоритет	Объем данных	Получатель
1	3 КБайт	А
0	20 КБайт	С
1	2 КБайт	С

Параметры сконструированной компьютерной сети:

1. Объем кадра = 4 Кб, включая 32 байта преамбулы и 256 байтов заголовка, то есть передаваемые данные составляют **не весь** объем кадра. Если данных недостаточно для заполнения кадра, то он дополняется двоичными 0.
2. Время передачи кадров на участках АВ – 4 сек, ВС – 6 сек, СА – 4 сек. Это время от начала передачи кадра передающим узлом до окончания приема кадра принимающим узлом.
3. Время, необходимое на анализ кадра на узле, вычисление времени удержания маркера и другие возможные задержки считаем незначительным и не учитываем при решении задачи.
4. Номинальное время обхода кольца (ВНО) установлено в 84 сек.

В момент включения сети узел С выпускает маркер (но не передает данных из своей очереди). **Определите, через сколько секунд после этого момента узел А закончит передавать все данные из своей очереди (получит подтверждение об успешной передаче последнего кадра из своей очереди данных).** В ответе укажите целое число секунд.

Вариант 2 Ответ: 358

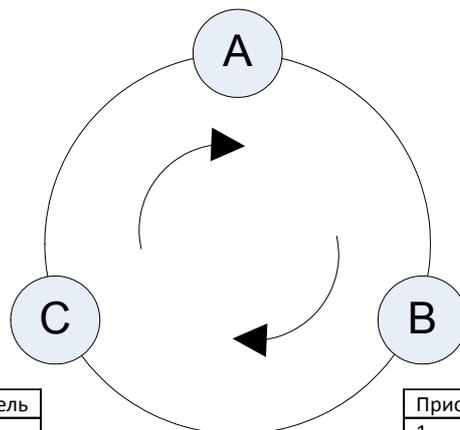
Студенты Иван и Петр сконструировали прототип компьютерной сети и реализовали стек протоколов, основывающийся на принципах стандарта IEEE 802.5.

Сеть имеет топологию кольцо (все узлы объединены последовательно в кольцо, а данные передаются по кругу в одном направлении, при этом каждый узел ретранслирует все проходящие кадры) и использует маркерный метод доступа. В сети циркулирует маркер (специальный кадр стандартного размера). Узел, получивший маркер, может начать передачу данных, для чего изымает маркер из сети и выпускает кадр данных, который передается последующими в кольце узлами, пока не попадет на узел-получатель, тот выставит в кадре признак приема и передаст кадр по кольцу далее, до узла отправителя. Тот в свою очередь изымет кадр из сети и либо освободит маркер, давая следующему по кольцу узлу начать передачу, либо продолжит удерживать маркер и начнет передачу следующего кадра из своей очереди, согласно представленным ниже правилам:

1. Узел будет удерживать маркер до тех пор, пока не опустеет его очередь или пока не истечет время удержания маркера данным узлом. Время удержания маркера узлом определяется как разность номинального времени обхода кольца (ВНО), задаваемого в виде константы в момент инициализации кольца, и времени реального отсутствия маркера на данном узле (ВРО). Время реального отсутствия маркера на узле определяется как интервал между моментом, когда маркер покинул данный узел в прошлый раз и моментом, когда узел принял маркер в следующий раз.
2. Имеется два типа трафика – с высшим приоритетом (1) и с обычным приоритетом (0). Признаком приоритета данные помечаются в очереди сетевого интерфейса, но приоритет влияет только на порядок передачи данных через сетевой интерфейс, порядок данных в очереди своего узла от приоритета не зависит и остается неизменным.
3. Когда узел получает маркер первый раз (от начала работы сети) – он может передать только один кадр (независимо от его приоритета), после чего обязан выпустить маркер дальше.
4. В общем случае, когда узел получает маркер, на узле вычисляется время удержания маркера. Если получившееся время (ВНО-ВРО) меньше или равно нулю, то узел передает только один кадр с высшим приоритетом, если такой стоит следующим в его очереди или пропускает маркер дальше. Если время удержания маркера на данном узле получается положительным, то узел начинает передачу данных из своей очереди по порядку и независимо от их приоритета.
5. Узел может продолжать передачу данных в течение времени удержания маркера, вычисленного в момент прихода маркера на узел, причем, если этот период истек до прихода очередного кадра с подтверждением приема, узел дожидается этого события, а потом выпускает маркер вне зависимости от приоритета передаваемых данных.

Схема сети и очереди сетевых интерфейсов представлены на рисунке (началу очередей соответствуют первые строки таблиц):

Приоритет	Объем данных	Получатель
0	1 КБайт	В
1	16 КБайт	В
0	9 КБайт	С



Приоритет	Объем данных	Получатель
0	1 КБайт	А
0	8 КБайт	В
1	2 КБайт	А

Приоритет	Объем данных	Получатель
1	3 КБайт	А
0	20 КБайт	С
1	2 КБайт	С

Параметры сконструированной компьютерной сети:

1. Объем кадра = 4 Кб, включая 32 байта преамбулы и 256 байтов заголовка, то есть передаваемые данные составляют **не весь** объем кадра. Если данных недостаточно для заполнения кадра, то он дополняется двоичными 0.

2. Время передачи кадров на участках АВ – 4 сек, ВС – 6 сек, СА – 4 сек. Это время от начала передачи кадра передающим узлом до окончания приема кадра принимающим узлом.
3. Время, необходимое на анализ кадра на узле, вычисление времени удержания маркера и другие возможные задержки считаем незначительным и не учитываем при решении задачи.
4. Номинальное время обхода кольца (ВНО) установлено в 84 сек.

В момент включения сети узел С выпускает маркер (но не передает данных из своей очереди). **Определите, через сколько секунд после этого момента узел В закончит передавать все данные из своей очереди (получит подтверждение об успешной передаче последнего кадра из своей очереди данных).** В ответе укажите целое число секунд.

Вариант 3 Ответ: 378

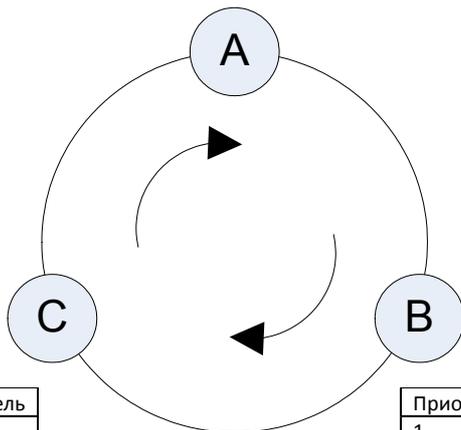
Студенты Иван и Петр сконструировали прототип компьютерной сети и реализовали стек протоколов, основывающийся на принципах стандарта IEEE 802.5.

Сеть имеет топологию кольцо (все узлы объединены последовательно в кольцо, а данные передаются по кругу в одном направлении, при этом каждый узел ретранслирует все проходящие кадры) и использует маркерный метод доступа. В сети циркулирует маркер (специальный кадр стандартного размера). Узел, получивший маркер, может начать передачу данных, для чего изымает маркер из сети и выпускает кадр данных, который передается последующими в кольце узлами, пока не попадет на узел-получатель, тот выставит в кадре признак приема и передаст кадр по кольцу далее, до узла отправителя. Тот в свою очередь изымет кадр из сети и либо освободит маркер, давая следующему по кольцу узлу начать передачу, либо продолжит удерживать маркер и начнет передачу следующего кадра из своей очереди, согласно представленным ниже правилам:

1. Узел будет удерживать маркер до тех пор, пока не опустеет его очередь или пока не истечет время удержания маркера данным узлом. Время удержания маркера узлом определяется как разность номинального времени обхода кольца (ВНО), задаваемого в виде константы в момент инициализации кольца, и времени реального отсутствия маркера на данном узле (ВРО). Время реального отсутствия маркера на узле определяется как интервал между моментом, когда маркер покинул данный узел в прошлый раз и моментом, когда узел принял маркер в следующий раз.
2. Имеется два типа трафика – с высшим приоритетом (1) и с обычным приоритетом (0). Признаком приоритета данные помечаются в очереди сетевого интерфейса, но приоритет влияет только на порядок передачи данных через сетевой интерфейс, порядок данных в очереди своего узла от приоритета не зависит и остается неизменным.
3. Когда узел получает маркер первый раз (от начала работы сети) – он может передать только один кадр (независимо от его приоритета), после чего обязан выпустить маркер дальше.
4. В общем случае, когда узел получает маркер, на узле вычисляется время удержания маркера. Если получившееся время (ВНО-ВРО) меньше или равно нулю, то узел передает только один кадр с высшим приоритетом, если такой стоит следующим в его очереди или пропускает маркер дальше. Если время удержания маркера на данном узле получается положительным, то узел начинает передачу данных из своей очереди по порядку и независимо от их приоритета.
5. Узел может продолжать передачу данных в течение времени удержания маркера, вычисленного в момент прихода маркера на узел, причем, если этот период истек до прихода очередного кадра с подтверждением приема, узел дожидается этого события, а потом выпускает маркер вне зависимости от приоритета передаваемых данных.

Схема сети и очереди сетевых интерфейсов представлены на рисунке (началу очередей соответствуют первые строки таблиц):

Приоритет	Объем данных	Получатель
0	1 КБайт	В
1	16 КБайт	В
0	9 КБайт	С



Приоритет	Объем данных	Получатель
0	1 КБайт	А
0	8 КБайт	В
1	2 КБайт	А

Приоритет	Объем данных	Получатель
1	3 КБайт	А
0	20 КБайт	С
1	2 КБайт	С

Параметры сконструированной компьютерной сети:

1. Объем кадра = 4 Кб, включая 32 байта преамбулы и 256 байтов заголовка, то есть передаваемые данные составляют **не весь** объем кадра. Если данных недостаточно для заполнения кадра, то он дополняется двоичными 0.
2. Время передачи кадров на участках АВ – 4 сек, ВС – 6 сек, СА – 4 сек. Это время от начала передачи кадра передающим узлом до окончания приема кадра принимающим узлом.
3. Время, необходимое на анализ кадра на узле, вычисление времени удержания маркера и другие возможные задержки считаем незначительным и не учитываем при решении задачи.
4. Номинальное время обхода кольца (ВНО) установлено в 84 сек.

В момент включения сети узел С выпускает маркер (но не передает данных из своей очереди). **Определите, через сколько секунд после этого момента узел С закончит передавать все данные из своей очереди (получит подтверждение об успешной передаче последнего кадра из своей очереди данных).** В ответе укажите целое число секунд.

Вариант 4 Ответ: 424

Студенты Иван и Петр сконструировали прототип компьютерной сети и реализовали стек протоколов, основывающийся на принципах стандарта IEEE 802.5.

Сеть имеет топологию кольца (все узлы объединены последовательно в кольцо, а данные передаются по кругу в одном направлении, при этом каждый узел ретранслирует все проходящие кадры) и использует маркерный метод доступа. В сети циркулирует маркер (специальный кадр стандартного размера). Узел, получивший маркер, может начать передачу данных, для чего изымает маркер из сети и выпускает кадр данных, который передается последующими в кольце узлами, пока не попадет на узел-получатель, тот выставит в кадре признак приема и передаст кадр по кольцу далее, до узла отправителя. Тот в свою очередь изымет кадр из сети и либо освободит маркер, давая следующему по кольцу узлу начать передачу, либо продолжит удерживать маркер и начнет передачу следующего кадра из своей очереди, согласно представленным ниже правилам:

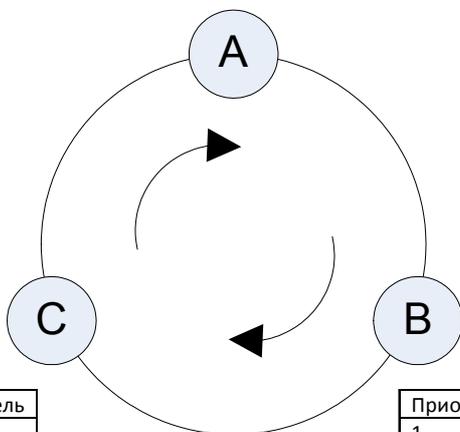
1. Узел будет удерживать маркер до тех пор, пока не опустеет его очередь или пока не истечет время удержания маркера данным узлом. Время удержания маркера узлом определяется как разность номинального времени обхода кольца (ВНО), задаваемого в виде константы в момент инициализации кольца, и времени реального отсутствия маркера на данном узле (ВРО). Время реального отсутствия маркера на узле определяется как интервал между моментом, когда маркер покинул данный узел в прошлый раз и моментом, когда узел принял маркер в следующий раз.
2. Имеется два типа трафика – с высшим приоритетом (1) и с обычным приоритетом (0). Признаком приоритета данные помечаются в очереди сетевого интерфейса, но приоритет влияет только на

порядок передачи данных через сетевой интерфейс, порядок данных в очереди своего узла от приоритета не зависит и остается неизменным.

3. Когда узел получает маркер первый раз (от начала работы сети) – он может передать только один кадр (независимо от его приоритета), после чего обязан выпустить маркер дальше.
4. В общем случае, когда узел получает маркер, на узле вычисляется время удержания маркера. Если получившееся время (ВНО-ВРО) меньше или равно нулю, то узел передает только один кадр с высшим приоритетом, если такой стоит следующим в его очереди или пропускает маркер дальше. Если время удержания маркера на данном узле получается положительным, то узел начинает передачу данных из своей очереди по порядку и независимо от их приоритета.
5. Узел может продолжать передачу данных в течение времени удержания маркера, вычисленного в момент прихода маркера на узел, причем, если этот период истек до прихода очередного кадра с подтверждением приема, узел дожидается этого события, а потом выпускает маркер вне зависимости от приоритета передаваемых данных.

Схема сети и очереди сетевых интерфейсов представлены на рисунке (началу очередей соответствуют первые строки таблиц):

Приоритет	Объем данных	Получатель
1	1 КБайт	С
0	16 КБайт	В
1	9 КБайт	В



Приоритет	Объем данных	Получатель
0	1 КБайт	А
0	8 КБайт	В
1	2 КБайт	А

Приоритет	Объем данных	Получатель
1	3 КБайт	С
0	14 КБайт	А
1	2 КБайт	А

Параметры сконструированной компьютерной сети:

1. Объем кадра = 4 Кб, включая 32 байта преамбулы и 256 байтов заголовка, то есть передаваемые данные составляют **не весь** объем кадра. Если данных недостаточно для заполнения кадра, то он дополняется двоичными 0.
2. Время передачи кадров на участках АВ – 4 сек, ВС – 6 сек, СА – 4 сек. Это время от начала передачи кадра передающим узлом до окончания приема кадра принимающим узлом.
3. Время, необходимое на анализ кадра на узле, вычисление времени удержания маркера и другие возможные задержки считаем незначительным и не учитываем при решении задачи.
4. Номинальное время обхода кольца (ВНО) установлено в 39 сек.

В момент включения сети узел С выпускает маркер (но не передает данных из своей очереди). **Определите, через сколько секунд после этого момента узел А закончит передавать все данные из своей очереди (получит подтверждение об успешной передаче последнего кадра из своей очереди данных).** В ответе укажите целое число секунд.

Вариант 5 Ответ: 246

Студенты Иван и Петр сконструировали прототип компьютерной сети и реализовали стек протоколов, основывающийся на принципах стандарта IEEE 802.5.

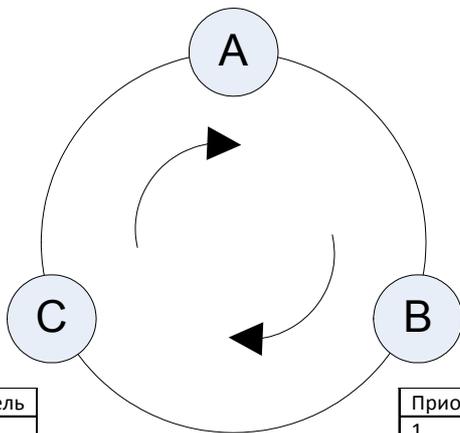
Сеть имеет топологию кольца (все узлы объединены последовательно в кольцо, а данные передаются по кругу в одном направлении, при этом каждый узел ретранслирует все проходящие кадры) и использует маркерный метод доступа. В сети циркулирует маркер (специальный кадр стандартного размера). Узел, получивший маркер, может начать передачу данных, для чего изымает маркер из сети и выпускает кадр

данных, который передается последующими в кольце узлами, пока не попадет на узел-получатель, тот выставит в кадре признак приема и передаст кадр по кольцу далее, до узла отправителя. Тот в свою очередь изымет кадр из сети и либо освободит маркер, давая следующему по кольцу узлу начать передачу, либо продолжит удерживать маркер и начнет передачу следующего кадра из своей очереди, согласно представленным ниже правилам:

1. Узел будет удерживать маркер до тех пор, пока не опустеет его очередь или пока не истечет время удержания маркера данным узлом. Время удержания маркера узлом определяется как разность номинального времени обхода кольца (ВНО), задаваемого в виде константы в момент инициализации кольца, и времени реального отсутствия маркера на данном узле (ВРО). Время реального отсутствия маркера на узле определяется как интервал между моментом, когда маркер покинул данный узел в прошлый раз и моментом, когда узел принял маркер в следующий раз.
2. Имеется два типа трафика – с высшим приоритетом (1) и с обычным приоритетом (0). Признаком приоритета данные помечаются в очереди сетевого интерфейса, но приоритет влияет только на порядок передачи данных через сетевой интерфейс, порядок данных в очереди своего узла от приоритета не зависит и остается неизменным.
3. Когда узел получает маркер первый раз (от начала работы сети) – он может передать только один кадр (независимо от его приоритета), после чего обязан выпустить маркер дальше.
4. В общем случае, когда узел получает маркер, на узле вычисляется время удержания маркера. Если получившееся время (ВНО-ВРО) меньше или равно нулю, то узел передает только один кадр с высшим приоритетом, если такой стоит следующим в его очереди или пропускает маркер дальше. Если время удержания маркера на данном узле получается положительным, то узел начинает передачу данных из своей очереди по порядку и независимо от их приоритета.
5. Узел может продолжать передачу данных в течение времени удержания маркера, вычисленного в момент прихода маркера на узел, причем, если этот период истек до прихода очередного кадра с подтверждением приема, узел дожидается этого события, а потом выпускает маркер вне зависимости от приоритета передаваемых данных.

Схема сети и очереди сетевых интерфейсов представлены на рисунке (началу очередей соответствуют первые строки таблиц):

Приоритет	Объем данных	Получатель
1	1 КБайт	С
0	16 КБайт	В
1	9 КБайт	В



Приоритет	Объем данных	Получатель
0	1 КБайт	А
0	8 КБайт	В
1	2 КБайт	А

Приоритет	Объем данных	Получатель
1	3 КБайт	С
0	14 КБайт	А
1	2 КБайт	А

Параметры сконструированной компьютерной сети:

1. Объем кадра = 4 Кб, включая 32 байта преамбулы и 256 байтов заголовка, то есть передаваемые данные составляют **не весь** объем кадра. Если данных недостаточно для заполнения кадра, то он дополняется двоичными 0.
2. Время передачи кадров на участках АВ – 4 сек, ВС – 6 сек, СА – 4 сек. Это время от начала передачи кадра передающим узлом до окончания приема кадра принимающим узлом.
3. Время, необходимое на анализ кадра на узле, вычисление времени удержания маркера и другие возможные задержки считаем незначительным и не учитываем при решении задачи.
4. Номинальное время обхода кольца (ВНО) установлено в 39 сек.

В момент включения сети узел С выпускает маркер (но не передает данных из своей очереди). **Определите, через сколько секунд после этого момента узел В закончит передавать все данные из своей очереди (получит подтверждение об успешной передаче последнего кадра из своей очереди данных).** В ответе укажите целое число секунд.

Вариант 6 Ответ: 224

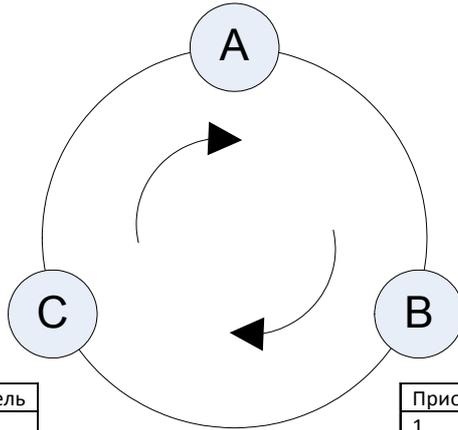
Студенты Иван и Петр сконструировали прототип компьютерной сети и реализовали стек протоколов, основывающийся на принципах стандарта IEEE 802.5.

Сеть имеет топологию кольцо (все узлы объединены последовательно в кольцо, а данные передаются по кругу в одном направлении, при этом каждый узел ретранслирует все проходящие кадры) и использует маркерный метод доступа. В сети циркулирует маркер (специальный кадр стандартного размера). Узел, получивший маркер, может начать передачу данных, для чего изымает маркер из сети и выпускает кадр данных, который передается последующими в кольце узлами, пока не попадет на узел-получатель, тот выставит в кадре признак приема и передаст кадр по кольцу далее, до узла отправителя. Тот в свою очередь изымет кадр из сети и либо освободит маркер, давая следующему по кольцу узлу начать передачу, либо продолжит удерживать маркер и начнет передачу следующего кадра из своей очереди, согласно представленным ниже правилам:

1. Узел будет удерживать маркер до тех пор, пока не опустеет его очередь или пока не истечет время удержания маркера данным узлом. Время удержания маркера узлом определяется как разность номинального времени обхода кольца (ВНО), задаваемого в виде константы в момент инициализации кольца, и времени реального отсутствия маркера на данном узле (ВРО). Время реального отсутствия маркера на узле определяется как интервал между моментом, когда маркер покинул данный узел в прошлый раз и моментом, когда узел принял маркер в следующий раз.
2. Имеется два типа трафика – с высшим приоритетом (1) и с обычным приоритетом (0). Признаком приоритета данные помечаются в очереди сетевого интерфейса, но приоритет влияет только на порядок передачи данных через сетевой интерфейс, порядок данных в очереди своего узла от приоритета не зависит и остается неизменным.
3. Когда узел получает маркер первый раз (от начала работы сети) – он может передать только один кадр (независимо от его приоритета), после чего обязан выпустить маркер дальше.
4. В общем случае, когда узел получает маркер, на узле вычисляется время удержания маркера. Если получившееся время (ВНО-ВРО) меньше или равно нулю, то узел передает только один кадр с высшим приоритетом, если такой стоит следующим в его очереди или пропускает маркер дальше. Если время удержания маркера на данном узле получается положительным, то узел начинает передачу данных из своей очереди по порядку и независимо от их приоритета.
5. Узел может продолжать передачу данных в течение времени удержания маркера, вычисленного в момент прихода маркера на узел, причем, если этот период истек до прихода очередного кадра с подтверждением приема, узел дожидается этого события, а потом выпускает маркер вне зависимости от приоритета передаваемых данных.

Схема сети и очереди сетевых интерфейсов представлены на рисунке (началу очередей соответствуют первые строки таблиц):

Приоритет	Объем данных	Получатель
1	1 КБайт	С
0	16 КБайт	В
1	9 КБайт	В



Приоритет	Объем данных	Получатель
0	1 КБайт	А
0	8 КБайт	В
1	2 КБайт	А

Приоритет	Объем данных	Получатель
1	3 КБайт	С
0	14 КБайт	А
1	2 КБайт	А

Параметры сконструированной компьютерной сети:

1. Объем кадра = 4 Кб, включая 32 байта преамбулы и 256 байтов заголовка, то есть передаваемые данные составляют **не весь** объем кадра. Если данных недостаточно для заполнения кадра, то он дополняется двоичными 0.
2. Время передачи кадров на участках АВ – 4 сек, ВС – 6 сек, СА – 4 сек. Это время от начала передачи кадра передающим узлом до окончания приема кадра принимающим узлом.
3. Время, необходимое на анализ кадра на узле, вычисление времени удержания маркера и другие возможные задержки считаем незначительным и не учитываем при решении задачи.
4. Номинальное время обхода кольца (ВНО) установлено в 39 сек.

В момент включения сети узел С выпускает маркер (но не передает данных из своей очереди). **Определите, через сколько секунд после этого момента узел С закончит передавать все данные из своей очереди (получит подтверждение об успешной передаче последнего кадра из своей очереди данных).** В ответе укажите целое число секунд.

9. Технологии обработки информации в электронных таблицах (1 балл)

Вариант 1 Ответ: 1 0 0

Дан фрагмент электронной таблицы, в которой ячейки диапазона A1:C8 заполнены числами 0 или 1 как показано на рисунке. В ячейку D1 записали формулу вида:

=ЕСЛИ(И(В1=X;ИЛИ(А1=Y; С1=Z));0;1), где вместо X, Y и Z были подставлены числа 0 или 1.

После этого ячейку D1 последовательно скопировали в ячейки диапазона D2:D8.

В результате получили следующие значения:

	A	B	C	D	E
1	0	0	0	1	
2	0	0	1	1	
3	0	1	0	0	
4	0	1	1	0	
5	1	0	0	1	
6	1	0	1	1	
7	1	1	0	0	
8	1	1	1	1	
9					

Определите, какие значения были подставлены вместо X, Y и Z. В ответ укажите через пробел сначала значение X, затем Y и затем Z.

Вариант 2 Ответ: 1 1 0

Дан фрагмент электронной таблицы, в которой ячейки диапазона A1:C8 заполнены числами 0 или 1 как показано на рисунке. В ячейку D1 записали формулу вида:

=ЕСЛИ(И(B1=X;ИЛИ(A1=Y; C1=Z));0;1), где вместо X, Y и Z были подставлены числа 0 или 1.

После этого ячейку D1 последовательно скопировали в ячейки диапазона D2:D8.

В результате получили следующие значения:

	A	B	C	D	E
1	0	0	0	1	
2	0	0	1	1	
3	0	1	0	0	
4	0	1	1	1	
5	1	0	0	1	
6	1	0	1	1	
7	1	1	0	0	
8	1	1	1	0	
9					

Определите, какие значения были подставлены вместо X, Y и Z. В ответ укажите через пробел сначала значение X, затем Y и затем Z.

Вариант 3 Ответ: 0 1 1

Дан фрагмент электронной таблицы, в которой ячейки диапазона A1:C8 заполнены числами 0 или 1 как показано на рисунке. В ячейку D1 записали формулу вида:

=ЕСЛИ(И(B1=X;ИЛИ(A1=Y; C1=Z));0;1), где вместо X, Y и Z были подставлены числа 0 или 1.

После этого ячейку D1 последовательно скопировали в ячейки диапазона D2:D8.

В результате получили следующие значения:

	A	B	C	D	E
1	0	0	0	1	
2	0	0	1	0	
3	0	1	0	1	
4	0	1	1	1	
5	1	0	0	0	
6	1	0	1	0	
7	1	1	0	1	
8	1	1	1	1	
9					

Определите, какие значения были подставлены вместо X, Y и Z. В ответ укажите через пробел сначала значение X, затем Y и затем Z.

Вариант 4 Ответ: 1 0 1 0

Дан фрагмент электронной таблицы, в которой ячейки диапазона A1:C8 заполнены числами 0 или 1 как показано на рисунке. В ячейку D1 записали формулу вида:

=ЕСЛИ(И(ИЛИ(B1=W; A1=X); ИЛИ(A1=Y; C1=Z));0;1), где вместо W, X, Y и Z были подставлены числа 0 или 1.

После этого ячейку D1 последовательно скопировали в ячейки диапазона D2:D8.

В результате получили следующие значения:

	A	B	C	D	E
1	0	0	0	0	
2	0	0	1	1	
3	0	1	0	0	
4	0	1	1	1	
5	1	0	0	1	
6	1	0	1	1	
7	1	1	0	0	
8	1	1	1	0	
9					

Определите, какие значения были подставлены вместо W, X, Y и Z. В ответ укажите через пробел сначала значение W, затем X, затем Y и затем Z.

Вариант 5 Ответ: 0 0 1 1

Дан фрагмент электронной таблицы, в которой ячейки диапазона A1:C8 заполнены числами 0 или 1 как показано на рисунке. В ячейку D1 записали формулу вида: =ЕСЛИ(И(ИЛИ(B1=W; A1=X); ИЛИ(A1=Y; C1=Z)); 0; 1), где вместо W, X, Y и Z были подставлены числа 0 или 1.

После этого ячейку D1 последовательно скопировали в ячейки диапазона D2:D8.

В результате получили следующие значения:

	A	B	C	D	E
1	0	0	0	1	
2	0	0	1	0	
3	0	1	0	1	
4	0	1	1	0	
5	1	0	0	0	
6	1	0	1	0	
7	1	1	0	1	
8	1	1	1	1	
9					

Определите, какие значения были подставлены вместо W, X, Y и Z. В ответ укажите через пробел сначала значение W, затем X, затем Y и затем Z.

Вариант 6 Ответ: 1 1 0 0

Дан фрагмент электронной таблицы, в которой ячейки диапазона A1:C8 заполнены числами 0 или 1 как показано на рисунке. В ячейку D1 записали формулу вида: =ЕСЛИ(И(ИЛИ(B1=W; A1=X); ИЛИ(A1=Y; C1=Z)); 0; 1), где вместо W, X, Y и Z были подставлены числа 0 или 1.

После этого ячейку D1 последовательно скопировали в ячейки диапазона D2:D8.

В результате получили следующие значения:

	A	B	C	D	E
1	0	0	0	1	
2	0	0	1	1	
3	0	1	0	0	
4	0	1	1	0	
5	1	0	0	0	
6	1	0	1	1	
7	1	1	0	0	
8	1	1	1	1	
9					

Определите, какие значения были подставлены вместо W, X, Y и Z. В ответ укажите через пробел сначала значение W, затем X, затем Y и затем Z.

10. Технологии хранения, поиска и сортировки информации (2 балла)

Вариант 1 Ответ: 7

В базе данных метеорологических измерений хранятся результаты единовременных замеров трех параметров (температура, влажность, давление) для различных моментов времени. Диапазоны допустимых значений для параметров следующие: температура – от 10,0 до 40,0 включительно, влажность – от 60,0 до 90,0 включительно, давление – от 740,0 до 780,0 включительно. Известно, что не было замеров, у которых значение хотя бы одного параметра с учетом точности измерения совпадало со значением этого параметра в другом замере. Известно количество записей, получаемых в ответ на ряд запросов к этой базе:

1. $Температура \leq 40,0 \text{ and } Влажность \leq 90,0 \text{ and } Давление \leq 780,0$ – 70 записей.
2. $Температура \leq 40,0 \text{ and } Влажность < 75,0 \text{ and } Давление \leq 780,0$ – 30 записей.
3. $Температура \leq 40,0 \text{ and } Влажность \leq 90,0 \text{ and } Давление \geq 760,0 \text{ and } Давление \leq 780,0$ – 40 записей.
4. $Температура < 25,0 \text{ and } Влажность \geq 75,0 \text{ and } Влажность \leq 90,0 \text{ and } Давление < 760,0$ – 8 записей.
5. $Температура \leq 40,0 \text{ and } Влажность < 75,0 \text{ and } Давление \geq 760,0 \text{ and } Давление \leq 780,0$ – 15 записей.

Сколько записей будет получено в ответ на запрос:

$Температура \geq 25,0 \text{ and } Температура \leq 40,0 \text{ and } Влажность \geq 75,0 \text{ and } Влажность \leq 90,0 \text{ and } Давление < 760,0$?

В ответе укажите целое число.

Вариант 2 Ответ: 3

В базе данных метеорологических измерений хранятся результаты единовременных замеров трех параметров (температура, влажность, давление) для различных моментов времени. Диапазоны допустимых значений для параметров следующие: температура – от 10,0 до 40,0 включительно, влажность – от 60,0 до 90,0 включительно, давление – от 740,0 до 780,0 включительно. Известно, что не было замеров, у которых значение хотя бы одного параметра с учетом точности измерения совпадало со значением этого параметра в другом замере. Известно количество записей, получаемых в ответ на ряд запросов к этой базе:

1. $Температура \leq 40,0 \text{ and } Влажность \leq 90,0 \text{ and } Давление \leq 780,0$ – 70 записей.
2. $Температура < 25,0 \text{ and } Влажность \leq 90,0 \text{ and } Давление \leq 780,0$ – 40 записей.
3. $Температура \leq 40,0 \text{ and } Влажность \leq 90,0 \text{ and } Давление < 760,0$ – 30 записей.
4. $Температура \geq 25,0 \text{ and } Температура \leq 40,0 \text{ and } Влажность \geq 75,0 \text{ and } Влажность \leq 90,0 \text{ and } Давление \geq 760,0 \text{ and } Давление \leq 780,0$ – 15 записей.
5. $Температура < 25,0 \text{ and } Влажность \leq 90,0 \text{ and } Давление < 760,0$ – 18 записей.

Сколько записей будет получено в ответ на запрос:

$Температура \geq 25,0 \text{ and } Температура \leq 40,0 \text{ and } Влажность < 75,0 \text{ and } Давление \geq 760,0 \text{ and } Давление \leq 780,0$?

В ответе укажите целое число.

Вариант 3 Ответ: 11

В базе данных метеорологических измерений хранятся результаты единовременных замеров трех параметров (температура, влажность, давление) для различных моментов времени. Диапазоны допустимых значений для параметров следующие: температура – от 10,0 до 40,0 включительно, влажность – от 60,0 до 90,0 включительно, давление – от 740,0 до 780,0 включительно. Известно, что не было замеров, у которых значение хотя бы одного параметра с учетом точности измерения совпадало со значением этого параметра в другом замере. Известно количество записей, получаемых в ответ на ряд запросов к этой базе:

1. $Температура \leq 40,0 \text{ and } Влажность \leq 90,0 \text{ and } Давление \leq 780,0$ – 70 записей.
2. $Температура \leq 40,0 \text{ and } Влажность \geq 75,0 \text{ and } Влажность \leq 90,0 \text{ and } Давление \leq 780,0$ – 40 записей.
3. $Температура \geq 25,0 \text{ and } Температура \leq 40,0 \text{ and } Влажность < 75,0 \text{ and } Давление \geq 760,0 \text{ and } Давление \leq 780,0$ – 4 записи.
4. $Температура \leq 40,0 \text{ and } Влажность \leq 90,0 \text{ and } Давление < 760,0$ – 30 записей.

5. *Температура* $\leq 40,0$ and *Влажность* $\geq 75,0$ and *Влажность* $\leq 90,0$ and *Давление* $\leq 760,0$ – 15 записей.

Сколько записей будет получено в ответ на запрос:

Температура $< 25,0$ and *Влажность* $< 75,0$ and *Давление* $\geq 760,0$ and *Давление* $\leq 780,0$?

В ответе укажите целое число.

Вариант 4 Ответ: 6

В базе данных метеорологических измерений хранятся результаты единовременных замеров трех параметров (температура, влажность, давление) для различных моментов времени. Диапазоны допустимых значений для параметров следующие: температура – от 10,0 до 40,0 включительно, влажность – от 60,0 до 90,0 включительно, давление – от 740,0 до 780,0 включительно. Известно, что не было замеров, у которых значение хотя бы одного параметра с учетом точности измерения совпадало со значением этого параметра в другом замере. Известно количество записей, получаемых в ответ на ряд запросов к этой базе:

1. *Температура* $\leq 40,0$ and (*Давление* $< 760,0$ and *Влажность* $\leq 90,0$ or *Давление* $\leq 780,0$ and *Влажность* $< 75,0$) – 35 записей.
2. *Температура* $\leq 40,0$ and *Влажность* $< 75,0$ and *Давление* $\leq 780,0$ – 22 записи.
3. *Температура* $\leq 40,0$ and *Влажность* $\geq 75,0$ and *Влажность* $\leq 90,0$ and *Давление* $\leq 780,0$ – 33 записи.
4. *Температура* $\leq 40,0$ and *Давление* $\geq 760,0$ and *Давление* $\leq 780,0$ and (*Температура* $\geq 25,0$ and *Влажность* $\leq 90,0$ or *Влажность* $\geq 75,0$ and *Влажность* $\leq 90,0$) – 26 записей.

Сколько записей будет получено в ответ на запрос:

Температура $\geq 25,0$ and *Температура* $\leq 40,0$ and *Влажность* $< 75,0$ and *Давление* $\geq 760,0$ and *Давление* $\leq 780,0$?

В ответе укажите целое число.

Вариант 5 Ответ: 3

В базе данных метеорологических измерений хранятся результаты единовременных замеров трех параметров (температура, влажность, давление) для различных моментов времени. Диапазоны допустимых значений для параметров следующие: температура – от 10,0 до 40,0 включительно, влажность – от 60,0 до 90,0 включительно, давление – от 740,0 до 780,0 включительно. Известно, что не было замеров, у которых значение хотя бы одного параметра с учетом точности измерения совпадало со значением этого параметра в другом замере. Известно количество записей, получаемых в ответ на ряд запросов к этой базе:

1. *Температура* $\leq 40,0$ and (*Давление* $\geq 760,0$ and *Давление* $\leq 780,0$ and *Влажность* $\leq 90,0$ or *Давление* $\leq 780,0$ and *Влажность* $\geq 75,0$ and *Влажность* $\leq 90,0$) – 43 записей.
2. *Температура* $\leq 40,0$ and *Влажность* $< 75,0$ and *Давление* $\leq 780,0$ – 22 записи.
3. *Температура* $\leq 40,0$ and *Влажность* $\geq 75,0$ and *Влажность* $\leq 90,0$ and *Давление* $\leq 780,0$ – 33 записи.
4. *Давление* $< 760,0$ and (*Влажность* $< 75,0$ and *Температура* $\leq 40,0$ or *Влажность* $\leq 90,0$ and *Температура* $< 25,0$) – 15 записей.

Сколько записей будет получено в ответ на запрос:

Температура $< 25,0$ and *Влажность* $\geq 75,0$ and *Влажность* $\leq 90,0$ and *Давление* $< 760,0$?

В ответе укажите целое число.

Вариант 6 Ответ: 12

В базе данных метеорологических измерений хранятся результаты единовременных замеров трех параметров (температура, влажность, давление) для различных моментов времени. Диапазоны допустимых значений для параметров следующие: температура – от 10,0 до 40,0 включительно, влажность – от 60,0 до 90,0 включительно, давление – от 740,0 до 780,0 включительно. Известно, что не было замеров, у которых значение хотя бы одного параметра с учетом точности измерения совпадало со значением этого параметра в другом замере. Известно количество записей, получаемых в ответ на ряд запросов к этой базе:

1. *Температура* $\leq 40,0$ and *Влажность* ≤ 90 and (*Давление* $< 760,0$ or *Давление* $\leq 780,0$ and *Влажность* $\geq 75,0$) – 45 записей.

2. Температура $\leq 40,0$ and Влажность $< 75,0$ and Давление $\leq 780,0$ – 22 записи.
3. Температура $\leq 40,0$ and Влажность $\geq 75,0$ and Влажность $\leq 90,0$ and Давление $\leq 780,0$ – 33 записи.
4. Давление $\geq 760,0$ and Давление $\leq 780,0$ and (Влажность $< 75,0$ and Температура $\leq 40,0$ or Влажность $\leq 90,0$ and Температура $< 25,0$) – 22 записей.

Сколько записей будет получено в ответ на запрос:

Температура $< 25,0$ and Влажность $\geq 75,0$ and Влажность $\leq 90,0$ and Давление $\geq 760,0$ and Давление $\leq 780,0$?

В ответе укажите целое число.

11. Технологии программирования (2 балла)

Вариант 1

В зале кинотеатра n рядов по m мест в каждом. Кинотеатр недавно открылся, и Вас попросили написать недостающую часть программного обеспечения для терминала продажи билетов. А именно, вам нужно написать программу для определения того, на какой ряд продать билет покупателю.

В начальный момент все места в зале свободны. Потом начинают подходить зрители. Зритель номер i хочет купить билет в a_i -м ряду. Если весь a_i -ряд занят, то покупатель может купить себе место и в другом ряду, но такое, чтобы расстояние от выбранного ряда до желаемого было как можно меньше. Если таких рядов два, то он выберет ряд с наибольшим номером. Например, если покупатель хочет билет на полностью занятый третий ряд, а второй и четвёртый свободны, он возьмёт билет на четвёртый.

Для каждого покупателя нужно выдать ряд, на который ему будет продан билет.

Формат входного файла

В первой строке входного файла `input.txt` заданы целые числа n и m ($1 \leq n, m \leq 100$) — количество рядов и мест в ряду, соответственно. Во второй строке задано целое число k — количество покупателей, пришедших за билетами ($1 \leq k \leq n * m$). Далее, в k строках заданы числа a_i — номера желаемых рядов ($1 \leq a_i \leq n$).

Формат выходного файла

Для каждого покупателя выведите номер ряда, на который ему будет продан билет. Все ответы разделите переводами строк.

input.txt	output.txt
3 2	3
4	2
3	2
2	3
2	
2	

Вариант 2

В зале кинотеатра n рядов по m мест в каждом. Кинотеатр недавно открылся, и Вас попросили написать недостающую часть программного обеспечения для терминала продажи билетов. А именно, Вам нужно написать программу для определения мест, на которые можно продать билеты.

В начальный момент все места в зале свободны. Потом начинают подходить группы зрителей. Группа номер i состоит из a_i человек, и они все хотят сесть на r_i -й ряд. Более того, все люди из группы хотят сидеть рядом. Например, группе из трёх человек можно выдать билеты на места 4, 5 и 6, но нельзя на 4, 5 и 7.

Для каждой группы вам нужно выдать, есть ли такое число свободных мест рядом на желаемом ряду. А также, если места есть, назвать наименьший номер места, начиная с которого можно посадить a_i человек. После этого группа покупает предложенные билеты, и соответствующие места становятся занятыми. Если свободных мест, соответствующих запросу, нет, то она не покупает билеты, и занятых мест не добавляется.

Формат входного файла

В первой строке входного файла **input.txt** заданы целые числа **n** и **m** ($1 \leq n, m \leq 100$) — количество рядов и мест в ряду соответственно. Во второй строке задано целое число **k** — количество групп, пришедших за билетами. Далее, в **k** строках заданы описания групп в формате **a_i r_i** ($1 \leq a_i \leq m, 1 \leq r_i \leq n$) — количество человек в группе и желаемый ряд соответственно.

Формат выходного файла

Для каждой группы выведите либо YES **b_i**, где **b_i** — наименьший номер места, начиная с которого посадили группу, если билеты продать можно, либо NO, если продать билеты на данный ряд нельзя. Ответы для групп разделите переводами строк.

input.txt	output.txt
3 3	YES 1
4	NO
2 1	YES 1
2 1	YES 3
2 2	
1 2	

Вариант 3

В аэропорту есть платная парковка для автомобилей, на которых приезжают встретить или проводить пассажиров. При въезде на парковку каждому водителю даётся карточка, и в систему записывается время въезда этого автомобиля на парковку. При выезде с парковки у водителя карточку забирают, и он оплачивает парковку в зависимости от времени, которое автомобиль на ней находился.

Время парковки рассчитывается по формуле $t = \text{out} - \text{in} + 1$, где **out** -- время выезда, а **in** -- время въезда, то есть оплачиваются и минута, в которую машина приехала, и минута, в которую она уехала.

Стоимость парковки рассчитывается согласно следующей таблице:

t	СТОИМОСТЬ
от 1 до 15 мин.	0 руб.
от 16 до 30 мин.	120 руб.
от 31 до 60 мин.	170 руб.
от 61 до 90 мин.	220 руб.
от 91 до 120 мин.	270 руб.
от 121 до 150 мин.	320 руб.
от 151 до 180 мин.	370 руб.
от 181 до 210 мин.	420 руб.
и т. д. 50 руб. за каждые 30 мин.	

Напишите программу, которая по данным с парковки определит, сколько всего денег заработала парковка на день.

Формат входного файла

В первой строке входного файла **input.txt** дано число **n** ($1 \leq n \leq 1000$) — количество записей с парковки. Далее, каждая в своей строке, идут сами записи. Каждая запись состоит из трех целых чисел **type_i t_i a_i**. **type_i** принимает значения 1 или -1. 1 соответствует приезду машины, -1 — отъезду машины. **t_i** — время в минутах, когда произошло это событие ($0 \leq t_i < 1440$). **a_i** — номер карточки, выданной соответствующему водителю ($1 \leq a_i \leq 1000$). Гарантируется, что каждый номер карточки

встречается ровно два раза, причём первый раз — с 1, второй — с -1. Записи идут в порядке неубывания времени.

Формат выходного файла

В выходной файл **output.txt** требуется вывести, сколько в сумме заплатили водители за день.

input.txt	output.txt
8	460
1 1 1	
-1 30 1	
1 100 100	
1 100 3	
-1 114 3	
-1 159 100	
1 160 2	
-1 190 2	

Вариант 4

В аэропорту есть платная парковка для автомобилей, на которых приезжают встретить или проводить пассажиров. При въезде на парковку каждому водителю даётся карточка, и в систему записывается время въезда этого автомобиля на парковку. При выезде с парковки у водителя карточку забирают, и он оплачивает парковку в зависимости от времени, которое автомобиль на ней находился.

Время парковки рассчитывается по формуле $t = \text{out} - \text{in} + 1$, где **out** — время выезда, а **in** — время въезда, то есть оплачиваются и минута, в которую машина приехала, и минута, в которую она уехала.

Стоимость парковки рассчитывается согласно следующей таблице:

t	СТОИМОСТЬ
от 1 до 15 мин.	0 руб.
от 16 до 30 мин.	120 руб.
от 31 до 60 мин.	170 руб.
от 61 до 90 мин.	220 руб.
от 91 до 120 мин.	270 руб.
от 121 до 150 мин.	320 руб.
от 151 до 180 мин.	370 руб.
от 181 до 210 мин.	420 руб.
и т. д. 50 руб. за каждые 30 мин.	

Формат входного файла

В первой строке входного файла **input.txt** дано число **n** ($1 \leq n \leq 1000$) — количество записей с парковки.

Далее, каждая в своей строке, идут сами записи. Каждая запись состоит из трех целых чисел **type_i**; **t_i**; **a_i**. **type_i** принимает значения 1 или -1. 1 соответствует приезду машины, -1 — отъезду машины. **t_i** — время в минутах, когда произошло это событие ($0 \leq t_i < 1440$). **a_i** — номер карточки, выданной соответствующему водителю ($1 \leq a_i \leq 1000$). Гарантируется, что каждый номер карточки встречается ровно два раза, причём первый раз — с 1, второй — с -1. Записи идут в порядке неубывания времени.

Формат выходного файла

В выходной файл **output.txt** требуется для каждого отъезжающего автомобиля в своей строке вывести, сколько рублей заплатил водитель. Выведите эти суммы в том порядке, в котором машины выезжали с парковки.

input.txt	output.txt
8	120
1 1 1	0
-1 30 1	170
1 100 100	170
1 100 3	
-1 114 3	
-1 159 100	
1 160 2	
-1 190 2	

12. Технологии программирования (2 балла)

Вариант 1

В банкомате есть купюры различных номиналов. После того, как человек отправляет запрос на выдачу ему некоторой суммы, программа, установленная в банкомате, определяет, сколько и каких купюр необходимо выдать этому человеку, чтобы выданная сумма была не больше запрошенной, но при этом максимально близка к ней. Происходит это следующим образом.

В первую очередь выдаются купюры самого большого номинала. Если эти купюры кончаются в банкомате, или если добавление еще одной купюры делает сумму больше запрашиваемой, то начинают выдаваться купюры номинала, второго по величине, и т. д. Так происходит до тех пор, пока выдаваемая сумма не будет в точности равна запрошенной, либо же пока не выяснится, что суммы не равны, но меньших купюр нет.

Вам необходимо разработать программу, обрабатывающую запросы к банкомату. Даны изначальные количества купюр каждого номинала и список запросов. Необходимо сообщить, какие суммы будут выданы по каждому запросу. При этом, естественно, нужно учитывать, что после каждого запроса число купюр в банкомате уменьшается.

Формат входного файла

В первой строке входного файла **input.txt** дано число n ($1 \leq n \leq 10$) — количество различных типов купюр. В следующих n строках входного файла описаны все возможные типы купюр. Каждый тип характеризуется двумя числами c_i и d_i ($1 \leq c_i \leq 5000$, $1 \leq d_i \leq 100$) — номинал купюры и количество купюр такого номинала. Типы купюр даны в порядке уменьшения номинала. Каждый следующий номинал является делителем предыдущего. В следующей строке дано число m ($1 \leq m \leq 100$) — количество поступивших запросов. В следующих m строках описаны запросы в порядке их поступления. Каждый запрос характеризуется одним числом x_i ($1 \leq x_i \leq 100000$) — запрашиваемой суммой.

Формат выходного файла

В выходной файл **output.txt** требуется вывести результаты обработки всех запросов в порядке поступления, по одному на строке. Про каждый запрос выведите сумму, выданную пользователю после обработки запроса.

input.txt	output.txt
3	
50 3	
10 20	
5 1	
3	
223	220
15	15
27	20

Вариант 2

В банкомате есть купюры различных номиналов. После того, как человек отправляет запрос на выдачу ему некоторой суммы, программа, установленная в банкомате, определяет, сколько и каких купюр необходимо выдать, и возможно ли в принципе выдать запрошенную сумму. Происходит это следующим образом.

В первую очередь выдаются купюры самого большого номинала. Если эти купюры кончаются в банкомате, или если добавление еще одной купюры делает сумму больше запрашиваемой, то начинают выдаваться купюры номинала, второго по величине, и т. д. Так происходит до тех пор, пока выдаваемая сумма не будет в точности равна запрошенной, либо же пока не выяснится, что суммы не равны, но меньших купюр нет.

Интересный факт, что если для любые двух существующих номиналов один из них нацело делится на второй, то такой алгоритм будет всегда работает оптимально.

Вам необходимо разработать программу, обрабатывающую запросы к банкомату. Даны изначальные количества купюр каждого номинала и список запросов. Необходимо сообщить, какие запросы будут удовлетворены, а какие — нет. При этом, естественно, нужно учитывать, что после каждого удовлетворенного запроса число купюр в банкомате уменьшается.

Формат входного файла

В первой строке входного файла **input.txt** дано число n ($1 \leq n \leq 10$) — количество различных типов купюр. В следующих n строках входного файла описаны все возможные типы купюр. Каждый тип характеризуется двумя числами c_i и d_i ($1 \leq c_i \leq 5000$, $1 \leq d_i \leq 100$) — номинал купюры и количество купюр такого номинала. Типы купюр даны в порядке уменьшения номинала. Каждый следующий номинал является делителем предыдущего. В следующей строке дано число m ($1 \leq m \leq 100$) — количество поступивших запросов. В следующих m строках описаны запросы в порядке их поступления. Каждый запрос характеризуется одним числом x_i ($1 \leq x_i \leq 100000$) — запрашиваемой суммой.

Формат выходного файла

В выходной файл **output.txt** требуется вывести результаты обработки всех запросов в порядке поступления, по одному на строке. Про удовлетворенные запросы выведите "YES", про неудовлетворенные — "NO".

input.txt	output.txt
3	YES
50 3	YES
10 20	NO
5 1	
3	
220	
15	
25	

Вариант 3

В некотором аэропорту установлена процедура обслуживания прилетающего самолета. В первую очередь самолет подается к терминалу прибытия самолетов, где его покидают пассажиры и из него выгружается багаж. После этого он должен переместиться в зону технического обслуживания, где он будет заправлен, осмотрен и, при необходимости, с ним будет произведен мелкий ремонт. После этого он подается к терминалу вылета, загружается багажом и пассажирами и снова взлетает. Все это необходимо делать с максимальной возможной скоростью, так как каждая минута пребывания самолета на земле дорого стоит.

Будем считать, что на разгрузку и загрузку самолета у терминалов уходит по a минут, на обслуживание -- b минут, а посадка, взлет и перемещения по аэропорту происходят мгновенно. Также будем считать, что у терминалов прибытия и вылета одновременно могут находиться по три самолета, а в зоне обслуживания -- два. Если самолет закончил какой-то этап процедуры, а на следующий этап попасть не может, так как там нет свободных мест, то он встает в очередь перед этим этапом, и попадает на него только тогда, когда туда попадут все самолеты, приземлившиеся до него и на этом этапе освободится еще одно место.

Соответственно, в какой-то момент перед некоторыми этапами обслуживания могут возникать очереди и из нескольких самолетов.

Вам известен график посадки самолетов в аэропорту. Вам необходимо сообщить суммарное время пребывания всех самолетов в аэропорту.

Формат входного файла

В первой строке входного файла **input.txt** дано число n ($1 \leq n \leq 20$) — количество самолетов, которые должны приземлиться. В следующей строке даны числа a и b ($1 \leq a, b \leq 50$) — время, необходимое на разгрузку/загрузку самолета и на его техническое обслуживание. В следующих n строках описаны прибывающие самолеты. Каждый описывается одним целым числом от 1 до 1000 — его временем прибытия. Самолеты перечислены в порядке прибытия (т.е. их времена прибытия возрастают).

Формат выходного файла

В выходной файл **output.txt** требуется вывести одно число — суммарное время пребывания всех самолетов в аэропорту.

input.txt	output.txt
5	
2 4	
1	
2	
3	
4	
5	48

Вариант 4

В некотором аэропорту установлена процедура обслуживания прилетающего самолета. В первую очередь самолет подается к терминалу прибытия самолетов, где его покидают пассажиры и из него выгружается багаж. После этого он должен переместиться в зону технического обслуживания, где он будет заправлен, осмотрен и, при необходимости, с ним будет произведен мелкий ремонт. После этого он подается к терминалу вылета, загружается багажом и пассажирами и снова взлетает. Все это необходимо делать с максимальной возможной скоростью, так как каждая минута пребывания самолета на земле дорого стоит.

Будем считать, что на разгрузку и загрузку самолета у терминалов уходит по a минут, на обслуживание — b минут, а посадка, взлет и перемещения по аэропорту происходят мгновенно. Также будем считать, что у терминалов прибытия и вылета одновременно могут находиться по два самолета, а в зоне обслуживания — три. Если самолет закончил какой-то этап процедуры, а на следующий этап попасть не может, так как там нет свободных мест, то он встает в очередь перед этим этапом, и попадает на него только тогда, когда туда попадут все самолеты, приземлившиеся до него и на этом этапе освободится еще одно место.

Соответственно, в какой-то момент перед некоторыми этапами обслуживания могут возникать очереди и из нескольких самолетов.

Вам известен график посадки самолетов в аэропорту. Вам необходимо сообщить, когда каждый из них снова окажется в воздухе.

Формат входного файла

В первой строке входного файла **input.txt** дано число n ($1 \leq n \leq 20$) — количество самолетов, которые должны приземлиться. В следующей строке даны числа a и b ($1 \leq a, b \leq 50$) — время, необходимое на разгрузку/загрузку самолета и на его техническое обслуживание. В следующих n строках описаны прибывающие самолеты. Каждый описывается одним целым числом от 1 до 1000 — его временем прибытия. Самолеты перечислены в порядке прибытия (т.е. их времена прибытия возрастают).

Формат выходного файла

В выходной файл **output.txt** требуется для каждого самолета вывести время, в которое он взлетит. Ответы для самолетов выводить в том же порядке, в котором самолеты перечислены в выходном файле. Ответы разделить переводами строк.

input.txt	output.txt
5	9

2 4	10
1	11
2	13
3	14
4	
5	