

РЕАЛЬНОСТЬ. ЗАДАЧА. АЛГОРИТМ – 2022

ИНФОРМАТИКА

1. Пользователь, перемещаясь из одного каталога в другой, последовательно посетил каталоги TASK, MATHS, USER, D:\, DOCS, PHOTOS. При каждом перемещении пользователь либо спускался в каталог на уровень ниже, либо поднимался на уровень выше. Каково полное имя каталога, из которого начал перемещение пользователь?

- 1) D:\DOCS\PHOTOS
- 2) D:\TASK\MATHS\USER
- 3) D:\USER\MATHS\TASK
- 4) D:\TASK\MATHS\USER\DOCS

Решение.

Так как пользователь последовательно посещал каталоги, то каталоги TASK и MATHS, MATHS и USER, USER и D:\, D:\ и DOCS, DOCS и PHOTOS – соседние. Данному условию удовлетворяет только ответ D:\USER\MATHS\TASK.

Ответ: 3

2. Маской подсети называется 32-разрядное двоичное число, которое определяет, какая часть IP-адреса компьютера относится к адресу сети, а какая часть IP-адреса определяет адрес компьютера в подсети. В маске подсети старшие биты, отведенные в IP-адресе компьютера для адреса сети, имеют значение 1; младшие биты, отведенные в IP-адресе компьютера для адреса компьютера в подсети, имеют значение 0.

Если маска подсети 255.255.255.240 и IP-адрес компьютера в сети 192.168.10.157, то порядковый номер компьютера в сети равен_____

Решение.

1. Так как первые три октета (октет - число маски, содержит 8 бит) все равны 255, то в двоичном виде они записываются как 24 единицы, а значит, первые три октета определяют адрес сети.

2. Запишем число 240 в двоичном виде.

11110000

3. Запишем последний октет IP-адреса компьютера в сети:

10011101

4. Сопоставим последний октет маски и адреса компьютера в сети:

1111**0000**

1001**1101**

Жирным выделена необходимая нам часть, отвечающая (по условию) за адрес компьютера в подсети. Переведем её в десятичную систему счисления:

$1101_2 = 13_{10}$.

Ответ: 13.

3. Исполнитель КУЗНЕЧИК живёт на числовой оси. Начальное положение КУЗНЕЧИКА – точка 0. Система команд Кузнечика:

Вперед 5 – Кузнечик прыгает вперёд (в положительном направлении числовой оси) на 5 единиц,

Назад 3 – Кузнечик прыгает назад на 3 единиц.

Какое наименьшее количество раз должна встретиться в программе команда «Назад 3», чтобы Кузнечик оказался в точке 18?

Решение.

Обозначим через x количество команд «Вперед 5» в программе, а через y – количество команд «Назад 3», причём x и y могут быть только неотрицательными целыми числами.

Для того, чтобы КУЗНЕЧИК попал в точку 18 из точки 0, должно выполняться условие:

$5x - 3y = 18$

Представим его в виде:

$$5x-18=3y$$

Из последнего уравнения видно, что левая часть должна делиться на 3.

Из всех решений нас интересует такое, при котором y – наименьшее возможное число.

Используем метод подбора:

$$x=1, y<0$$

$$x=2, y<0$$

$$x=3, y<0$$

$$x=4, y=2/3$$

$$x=5, y=7/3$$

$$x=6, y=12/3$$

Наименьшее число команд «Назад 3» $y=4$.

Ответ: 4.

4. В электронной таблице значение формулы =СРЗНАЧ(A1:A4) равно 13. Чему будет равно значение формулы =СУММ(A1:A3), если в ячейке A4 записано значение 15?

Решение.

Функция СРЗНАЧ(A1:A4) считает среднее арифметическое диапазона A1:A4, т. е. сумму значений четырёх ячеек A1, A2, A3, A4, делённую на 4. Умножим среднее значение на число ячеек и получим сумму значений ячеек $A1 + A2 + A3 + A4 = 13 * 4 = 52$.

Теперь, вычтем значение ячейки A4 и найдём сумму СУММ(A1:A3):
 $A1 + A2 + A3 = 52 - 15 = 37$.

Ответ: 37

5. В таблице приведены запросы и соответствующее количество страниц, которые нашёл по ним поисковый сервер в некотором сегменте Интернета:

Запрос	Количество страниц (в тыс.)
уравнение неравенство	10500
уравнение	7500
уравнение & неравенство	4000

Сколько страниц (в тысячах) найдёт данный поисковый сервер по запросу «неравенство»?

Варианты ответов:

- 1) 3500
- 2) 3000
- 3) 7500
- 4) 7000

Решение.

Построим диаграммы Эйлера-Венна.



Запросу «уравнение & неравенство» соответствует пересечение (4000 тыс. страниц), а запросу «уравнение» – весь левый круг (7500 тыс. страниц).

Тогда левый «обрезанный круг» – это $7500 - 4000 = 3500$.



Запросу «уравнение | неравенство» соответствуют оба круга (10500), тогда правый «обрезанный круг» – это $10500 - 7500 = 3000$.



Итак, мы посчитали количество страниц для каждой области:



Несложно увидеть, что по запросу «неравенство» будет найдено $3000 + 4000 = 7000$ (тыс. страниц).

Ответ: 4

6. Какое число будет напечатано в результате работы следующей программы? Для Вашего удобства программа приведена на пяти языках программирования.

Бейсик	Python	Паскаль
<pre> DIM N, S AS INTEGER N = 30 S = 5 WHILE S < 500 S = S * 2 N = N + 10 WEND PRINT N </pre>	<pre> n = 30 s = 5 while s < 500: s *= 2 n += 10 print(n) </pre>	<pre> var n, s: integer; begin n := 30; s := 5; while s < 500 do begin s := s * 2; n := n + 10; end; writeln(n); end. </pre>
C++		Алгоритмический язык
<pre> #include <iostream> using namespace std; int main() { int n, s; n = 30; s = 5; while (s < 500) { s = s * 2; n = n + 10; } cout << n << endl; } </pre>		<pre> алг нач цел n, s n := 30 s := 5 нц пока s < 500 s := s * 2 n := n + 10 кц вывод n кон </pre>

```
return 0;
}
```

Решение.

На каждом шаге алгоритма значение переменной S удваивается, а к значению N прибавляется 10. Построим трассировочную таблицу:

№ итерации	s < 500	S	N
0		5	30
1	true	10	40
2	true	20	50
3	true	40	60
4	true	80	70
5	true	160	80
6	true	320	90
7	true	640	100
8	false		

Ответ: 100

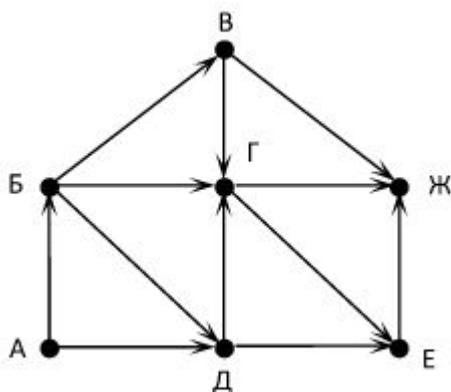
7. Сколько сообщений мог бы передавать светофор, если бы у него одновременно горели сразу три «глаза», а каждый из них мог бы менять цвет и становиться либо красным, либо желтым, либо зеленым?

Решение.

Один цвет по сути есть символ, а всё сообщение есть просто 3-буквенное слово. Из M = 3 различных символов можно составить Q = M^N слов длиной N = 3, т. е. 3³ = 27 слов.

Ответ: 27

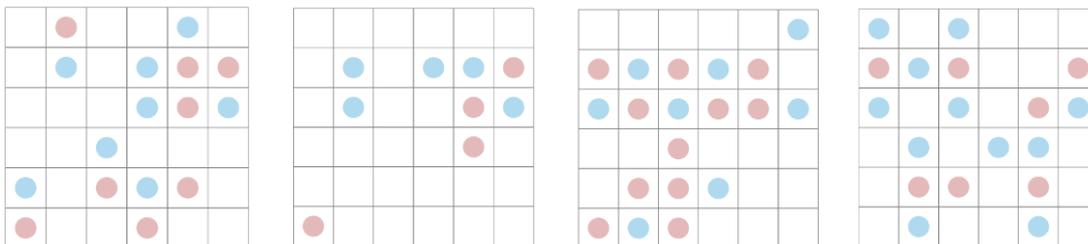
8. На рисунке – схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город Ж?



Решение.

Каждой вершине, начиная с начальной (А), поставим в соответствие индекс, равный количеству путей, которыми можно попасть в эту вершину. Для вершины А (начало пути) индекс всегда равен 1 (в начало пути можно попасть единственным образом – никуда не двигаясь). Теперь сформулируем правило: индекс вершины равен сумме индексов его предков. Исходя из этого индекс Б равен 1 (предок у Б один – вершина А).

У вершины Д предками являются А и Б, значит индекс вершины Д равен 1+1=2.



Решение.

Только для доски под буквой с все соседи каждого синего – красные, а среди соседей каждого красного есть хотя бы один синий.

Ответ: 3.

11.

Двоичная кодировка. Сообщение состоит только из букв А, В, С и D. Мы кодируем его по таким правилам.

- Каждую букву А заменяем последовательностью 101.
- Каждую букву В заменяем последовательностью 1100.
- Каждую букву С заменяем последовательностью 0110.
- Каждую букву D заменяем последовательностью 110.

В итоге получаем 11001100110110101110010110111000110110110110110. Сколько букв было в исходном сообщении?

Решение.

Совсем не обязательно декодировать полученную последовательность из нулей и единиц, а достаточно заметить, что код каждой буквы содержит ровно две единицы.

Поэтому количество букв в исходном сообщении равно половине количества единиц в закодированном сообщении, то есть равно 14.

Ответ: 14

12. Шоколадка с изюмом. Шоколадка состоит из одного ряда кубиков. Если в кубике есть изюминка, то обозначим его единицей, а иначе нулём. Сколько существует различных способов разломать шоколадку 00001110011000101100001110 на куски так, чтобы в любом была ровно одна изюминка.

Например, шоколадку 0010101 можно разломать четырьмя способами, а именно 001-01-01, 001-010-1, 0010-1-01 и 0010-10-1.

Решение.

Отбросим от шоколадки начальные и завершающие нули. Между ними всё равно нельзя её ломать и их исключение не повлияет на ответ. Получим 111001100010110000111.

Пока идут единицы, отламываем их по одной. Последнюю единицу перед группой нулей можно просто отломить, а также отломить с любым количеством нулей в группе.

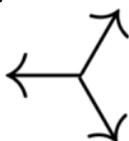
Пусть в первой группе k_1 нулей, то есть внутри её шоколадку можно разломать k_1+1 способами. Во второй группе k_2 нулей, то есть внутри её шоколадку можно разломать k_2+1 способами. И так далее. Тогда общее количество различных способов разломать шоколадку будет равно $(k_1+1)*(k_2+1)*...$

Для нашей конкретной шоколадки ответ равен $(2+1)*(3+1)*(1+1)*(4+1)=120$.

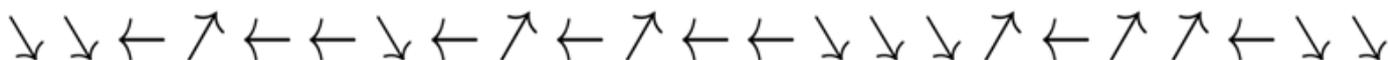
Ответ: 120.

13. Возвращение робота. Робот может выполнять три команды.

- Переехать на 1 метр влево по горизонтали.
- Переехать на 1 метр вверх под углом 60 градусов.
- Переехать на 1 метр вниз под углом 60 градусов.



Робот выполнил такую серию команд.



Какое минимальное число команд ему нужно выполнить дополнительно, чтобы вернуться в стартовую позицию?

Решение.

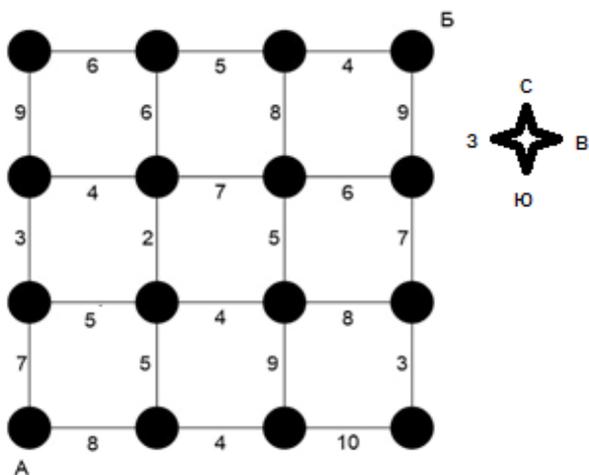
Одна команда «вверх», одна команд «вниз» и одна команд «влево» возвращают робота в стартовую позицию. Вычеркнем из серии команд такие тройки команд.



Серия упростилась до 3 команд «влево» и 2 команд «вниз». Чтобы робот вернулся в стартовую позицию, её необходимо дополнить одной командой «вниз» и 3 командами «вверх», то есть 4 командами.

Ответ: 4

14. Ученик Вася ходит в школу пешком. Он измерил время (минуты), за которое получается пройти каждый участок дороги и нанес на схему дороги. Линиями на схеме изображены дороги, точками – перекрестки. Помогите Васе понять, за какое минимальное количество минут он может попасть из дома (пункт А) в школу (пункт Б), если он ходит только на север или восток.



Решение.

Задача решается как стандартная задача о черепашке с помощью динамического программирования.

15	9	4	0
19	15	12	9
22	17	17	16
29	22	26	19

Ответ: 29

15. Определите, при каких наименьшем и наибольшем введённых значениях переменной s программа выведет число 32. В ответ запишите эти два числа по возрастанию через пробел. Для Вашего удобства программа представлена на нескольких языках программирования.

Бейсик	Паскаль	C++
<pre> DIM N, S AS INTEGER INPUT s s = s \ 10 n = 1 WHILE S < 39 s = s + 3 n = n * 2 WEND PRINT n </pre>	<pre> var s, n: integer; begin readln (s); s := s div 10; n := 1; while s < 39 do begin s := s + 3; n := n * 2 end; writeln(n) end. </pre>	<pre> #include <iostream> using namespace std; int main() { int s, n; cin >> s; s = s / 10; n = 1 ; while (s < 39) { s = s + 3; n = n * 2; } cout << n << endl; return 0; } </pre>
Python	Алгоритмический язык	
<pre> s = int(input()) s = s // 10 n = 1 while s < 39: s = s + 3 n = n * 2 print(n) </pre>	<pre> алг нач цел n, s ввод s s := div(s, 10) n := 1 нц пока s < 39 s := s + 3 n := n * 2 кц вывод n кон </pre>	

Решение.

Переменная n примет значение 32, если программа зайдет в тело цикла 5 раз. При этом переменная s увеличится на 15. Минимальное значение s, с которым нужно зайти первый раз в цикл – 24 (при выходе из цикла будет равно 24+15=39), максимальное – 26 (при выходе из цикла будет равно 26+15=41, при предыдущей итерации 26+12=38, следовательно будет выполнен еще один заход в тело цикла).

Исходно от числа отбрасывается последняя цифра. Из 24 сформируем минимальное число, полученное приписыванием одной цифры – 240, из 26 сформируем максимальное число, полученное приписыванием одной цифры – 269.

Ответ: 240 269

16. Замкнутый поезд. Поезд стоит на кольцевой линии. Его первый вагон сцеплен с последним. В случайных вагонах горит свет. Вы находитесь где-то в поезде и можете только переходить из одного вагона в другой и включить или выключать свет в них. Как посчитать количество вагонов в таком поезде?

Возможный алгоритм. Включаем свет в начальном вагоне, в котором мы находимся, если он ещё не горит. Затем идём вправо до тех пор, пока не встретим освещённый вагон. При этом считаем пройденные вагоны. Выключаем в найденном вагоне свет и идём влево к началному. Если в нём свет ещё горит, то повторяем путешествие вправо и обратно. Если же в нём свет не горит, то мы прошли полный круг и посчитали, сколько вагонов всего.

Обозначим единицей вагон с горящим светом, а нулём – без света. Пусть замкнутый поезд выглядит так 0000**111**001001000000, а вы находитесь в первом вагоне.

Посчитайте, сколько переходов из одного вагона в другой вам потребуется, чтобы посчитать количество вагонов по указанному алгоритму.

Решение.

Зажигаем свет в начальном вагоне.

1000111001001000000.

Бежим до первой единицы, обнуляем её и возвращаемся обратно. Делаем $4+4=8$ переходов из одного вагона в другой.

1000011001001000000.

Бежим до второй единицы. Делаем $5+5=10$ переходов.

1000001001001000000.

Бежим до третьей единицы. Делаем $6+6=12$ переходов.

1000000001001000000.

Бежим до четвёртой единицы. Делаем $9+9=18$ переходов.

1000000000001000000.

Бежим до пятой единицы. Делаем $12+12=24$ переходов.

1000000000000000000.

Бежим до последней единицы, совершая 19 переходов, обнуляем её, возвращаемся на 19 шагов обратно и видим, что в начальном вагоне света нет. Значит, в поезде всего 19 вагонов. На последнем этапе мы сделали $19+19=38$ переходов.

0000000000000000000.

Итого, чтобы посчитать количество вагонов, нам потребовалось $8+10+12+18+24+38=110$ переходов.

Ответ: 110

17. У исполнителя, который работает с положительными однобайтовыми двоичными числами, две команды, которым присвоены номера:

1. сдвинь влево
2. вычти 1

Выполняя первую из них, исполнитель сдвигает число на один двоичный разряд влево, причём на место освободившегося бита ставится 0. Выполняя вторую команду исполнитель вычитает из числа 1. Исполнитель начал вычисления с числа 120 и выполнил цепочку команд 121121. Запишите результат в десятичной системе.

Решение.

Если в старшем разряде двоичного числа нет единицы, то команда 1 удваивает число, если единица есть (т.е. десятичное число не меньше 128), то выводится остаток от деления удвоенного числа на 256. Таким образом, получим следующее:

- 1: 120 => 240,
- 2: 240 => 239,
- 1: 239 => 222 (остаток от $478 / 256$),
- 1: 222 => 188 (остаток от $444 / 256$),
- 2: 188 => 187,
- 1: 187 => 118 (остаток от $374 / 256$),

Ответ: 118.

Примечание.

Заметим, что в соответствии с условием программа работает с однобайтовыми числами. Если в старшем разряде числа стоит единица, то при сдвиге числа влево эта единица выходит за разрядную сетку и теряется. В результате при применении операции сдвига к числу, превышающему 127, то есть имеющему единицу в старшем разряде, получается остаток от деления данного числа на 256.

Ответ: 118.

18. Ниже на пяти языках программирования записана рекурсивная функция (процедура) F. Сколько чисел, не равных нулю, будет выведено на экран в результате вызова функции F(5)?

Бейсик	Python	C++
<pre> PROCEDURE F(n) PRINT n+1 IF n > 1 THEN F(n - 1) F(n - 3) END IF END PROCEDURE </pre>	<pre> def F(n): print(n+1) if n > 1: F(n - 1) F(n - 3) </pre>	<pre> void F(int n) { cout << n + 1; if (n > 1) { F(n - 1); F(n - 3); } } </pre>
Паскаль	Алгоритмический язык	
<pre> Procedure F(n: integer); begin writeln(n+1); if n > 1 then begin F(n - 1); F(n - 3); end; end; </pre>	<pre> алг F(цел n) нач вывод n + 1 если n > 1 то F(n - 1) F(n - 3) все кон </pre>	

Решение.

Промоделируем работу программы:

```

F(5): 6
  F(4): 5
    F(3): 4
      F(2): 3
        F(1): 2
          F(-1): 0
        F(0): 1
      F(1): 2
    F(2): 3
      F(1): 2
      F(-1): 0

```

Ответ: 9.

19. Падающие муравьи. По шесту гуляют муравьи. Их скорость равна одному сантиметру в секунду. Если муравей доходит до края шеста, то падает с него. Когда двое встречаются, они разворачиваются и ползут в противоположных направлениях. Пусть длина шеста равна 200 сантиметрам, муравьёв – 10 и для каждого указано, за сколько сантиметров от левого конца шеста он сидит и куда ползёт, влево или вправо.

Номер муравья	Расстояние от левого конца шеста	Направление движения
1	5	влево
2	10	влево
3	30	вправо
4	45	влево
5	66	вправо
6	77	влево
7	100	вправо
8	101	влево
9	150	влево
10	160	вправо

Через какое минимальное количество секунд все муравьи упадут с шеста?

Решение.

Все муравьи для нас одинаковы. Поэтому можно считать, что они не разворачиваются при встрече, а проходят сквозь друг друга. Без этого предположения решение было бы сложным и медленным. С ним всё просто.

Вычисляем самого левого муравья, ползущего направо. Это муравей №3. Он находится за 170 сантиметров до правого конца шеста и будет дольше всех туда добираться. Остальные муравьи, ползущие направо, упадут с шеста раньше.

Определяем самого правого муравья, ползущего налево. Это муравей №9. Он находится за 150 сантиметров до левого конца шеста и будет дольше всех туда добираться. Остальные муравьи, ползущие налево, упадут с шеста раньше.

Итак, эти два муравья упадут с шеста самыми последними, но самым последним будет №3. Ему потребуется 170 секунд. Именно через столько секунд шест окажется пустым.

Ответ: 170.

20. Чемпионат. На чемпионат по футболу приехало n команд. Они делятся на пары. Победившие в этих парах продолжают участвовать в чемпионате. Проигравшие отправляются домой. И так, пока не останется одна команда. Если какой-то команде не достаётся пары, то она просто переходит на следующий этап. Пусть $n=1000000$. Сколько игр будет в чемпионате?

Решение.

Сама формулировка задачи толкает к тому, чтобы провести все этапы чемпионата, считать, сколько игр прошло на каждом и добавлять к общему количеству, считать и добавлять.

Но если подумать, то считать не придётся. Вначале команд n . Чтобы в конце осталась одна, нужно исключить $(n-1)$ -у. Каждая игра всегда исключает одного участника и других способов исключения не бывает, поэтому число игр и число исключений равны. Ответ – в чемпионате будет $n-1$ игра.

Ответ: 999999.