Олимпиада школьников «Надежда энергетики». Заключительный этап. Очная форма.

ЗАДАНИЕ ПО ИНФОРМАТИКЕ
ВАРИАНТ 73111 для 11 класса

Для заданий 2, 3, 4, 5 требуется разработать алгоритм на языке блок-схем, псевдокоде или естественном языке

1. Утверждения $A \to C$, $A \;\&\; B \to D$, $\neg B \to E$ истинны. Чему равны A и B, если C, D и E ложны?

Схема решения. Таблица истинности для логической функции «импликация» представлена ниже.

X	Y	$X \to Y$
ложь	ложь	истина
ложь	истина	истина
истина	ложь	ложь
истина	истина	истина

Из таблицы видно, что если следствие ложно, то для того, чтобы вся формула была истинной, необходимо, чтобы посылка также была ложна. Таким образом, утверждение A должно быть ложно, в этом случае $A \;\&\; B$ также будет ложно. Кроме того, ложно должно быть $\neg B$, т.е. утверждение B должно быть истинно.

2. В археологических раскопках в Крыму при строительстве трассы «Таврида» археологи нашли табличку с таким текстом: $\sqrt{19} = 4 + \cfrac{1}{2 + \cfrac{1}{1 + \cfrac{1}{3 + \cfrac{1}{1 + \cfrac{1}{2 + \cfrac{1}{8 + \ldots}}}}}}$

   Далее в формуле структура из $2, 1, 3, 1, 2, 8$ повторяется бесконечное число раз. Пожалуйста, проверьте записанное предположение – разработайте алгоритм проверки на ЭВМ
   с точностью до 0.0001 справедливости этой формулы.

Схема решения. Данную дробь можно вычислить по формуле

$$f_1 = 1 / (2 + 1 / (1 + 1 / (3 + 1 / (1 + 1 / (2 + 1 / (8 + f_0)))))),$$

где f_0 – отбрасываемая часть, обозначенная многоточием. Положим сначала f_0 равной 0 и вычислим f_1. Если разность по модулю между переменными f_0 и f_1 окажется меньше 0.0001, можно прекращать вычисления. Иначе положим переменную f_0 равной f_1 и снова вычислим f_1 по той же формуле. Итоговый результат равен $4 + f_1$.

3. По прямоугольной области вертикальной стены размера $M \times N$ плит прыжками передвигается робот. Изначально робот находится в левом нижнем углу (Н) области. Робот может передвигаться, прыгая одним из четырех способов, показанных на рисунке. В таблице размера $M \times N$ указано, на каких плитах области находятся ступеньки. Ступенька всегда есть на левой нижней и правой верхней плитах. Если ступенька находится на вертикальном участке траектории движения робота, то совершать прыжок нельзя (робот разбивается). Если ступенька находится на горизонтальном участке траектории движения робота, то совершать прыжок можно. Приземление возможно только на ступеньку. При этом приземление возможно на первую встреченную ступеньку, находящуюся ниже плиты, завершающей траекторию, если на ней ступеньки нет (робот падает вниз, не разбиваясь). Если же в области под плитой, завершающей траекторию, ступенек нет, то робот разбивается. Разработайте алгоритм, отвечающий на вопрос: может ли робот добраться до указанной верхней правой плиты (К). Выход за пределы области не является возможным (робот разбивается).

Траектории движения робота

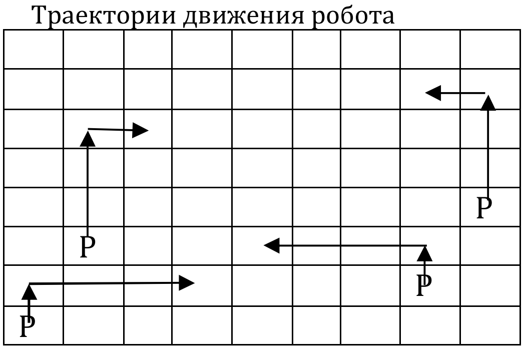

Схема зала (колонны расставлены условно)

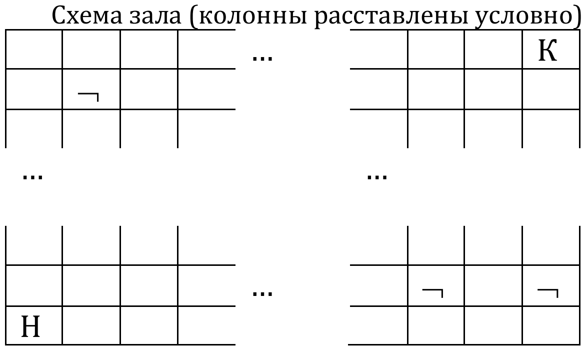

Схема решения. Из каждой точки робот может, в принципе, совершить четыре разных прыжка. Однако, некоторые прыжки могут закончиться падением, и робот разобьётся. Поэтому на каждом шаге надо проверять, куда именно может прыгнуть робот.

Запишем в массив координаты начальной точки. Далее в цикле извлекаем (удаляем) из массива координаты первой находящейся в нём точки, взамен кладём в него координаты точек, куда робот может прыгнуть из этой точки. Таких точек может оказаться от 0 до 4. Если среди них есть конечная точка, значит, робот может туда добраться. Если же робот в принципе не может добраться до конечной точки, то в какой-то момент массив станет пустым, и можно будет прекратить поиск.

4. Хранитель леса снова решил поиграть и окутал дорогу туманом, превратив в лабиринт. На входе разместил схему лабиринта (квадратная таблица размером $N \times N$). Стрелками обозначены проходы, по которым можно идти, не опасаясь быть пойманным местными энтами. На каждом шаге при движении по лабиринту встречаются шишки с вырезанными на них натуральными числами, которые можно собирать. Путник Ткач Туманов собирал шишки и попался на шутку Хранителя леса. Помогите Ткачу Туманов пройти по лабиринту из левого нижнего угла в правый верхний и собрать шишки, на которых записаны простые числа. Число называется простым, если оно делится только на само себя и на 1.

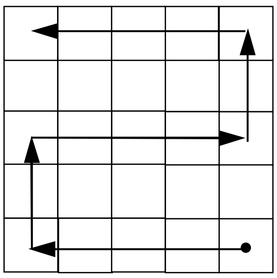

Схема решения. При такой схеме лабиринта проход из левого нижнего угла в правый верхний возможен, только если количество строк N выражается формулой $4K + 1$, где K – любое число от 0. Поэтому если $N - 1$ не делится нацело на 4, то проход не возможен. Иначе вычисляем K, это будет количество циклов. В каждом цикле идём из по текущей строке i (i начинается с N) слева направо, затем обрабатываем ячейку с индексами ($i - 1, N$), затем идём по строке $i - 2$ справа налево, и, наконец, обрабатываем ячейку с индексами ($i - 3, 1$). Строка с номером 1 проходится слева направо после завершения обработки остальных строк. При проходе проверяем числа в обрабатываемых ячеек на простоту. Если число является простым, выводим его. Можно сохранять найденные простые числа в массив, а потом вывести этот массив.

Чтобы проверить, что число k является простым, необходимо перебрать его возможные делители от 2 до $\sqrt{k}$ и проверить, делится ли число k на какой-либо из возможных делителей. Если это так, то число не является простым. При этом число 2 является простым, а число 1 не является простым.

Для сокращения перебора можно сначала проверить делимость числа k на 2, а потом проверить его делимость на возможные нечётные делители от 3 до $\sqrt{k}$ с шагом 2. Не забываем, что само число 2 является простым.

Можно ещё больше сократить перебор, отбрасывая числа, которые делятся на 2 и на 3. Для этого для числа k проверяются возможные делители i от 5 до $\sqrt{n}$ с шагом 6, но на каждом шаге цикла проверяется делимость числа k на i и на $(i + 2)$, т.е. получается ряд 5, 7, 11, 13 и т.д. До цикла надо проверить делимость числа n на 2 и на 3, а также надо учесть, что сами числа 2 и 3 являются простыми.

5. В таблице размером $N \times 2$ записаны координаты точек на плоскости (x, y). N достаточно велико. Все точки лежат на графике некоторой функции. Таким образом, функция задана табличным способом. Разработайте алгоритм проверки того, что функция является монотонно возрастающей.

Схема решения. Отсортируем точки по x-координате. Учитывая, что N велико, необходимо использовать эффективные алгоритмы сортировки. После этого проверим, что y-координата каждой точки, начиная со второй, не меньше y-координаты предыдущей точки.