|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Тренировочный вариант № 6 (2 часть).   |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | **24** | (№ 492) Дано целое положительное число N. Необходимо определить наименьшее целое число K, для которого выполняется неравенство:  1 + 2 + ... + K > N.  Программист написал программу неправильно.   |  |  |  | | --- | --- | --- | | Паскаль | Python | Си | | var n, k: integer;  begin  read(n);   k := 1;  while n > 0 do begin   n := n - k;   k := k + 1;   end;  writeln(k)  end. | n = int(input()) k = 1 while n > 0:  n = n - k  k = k + 1 print(k) | #include <stdio.h> int main() {  int n, k;  scanf("%d",&n);  k = 1;  while (n > 0) {  n = n - k;  k = k + 1;  }  printf("%d", k);  return 0; } |   Последовательно выполните следующее.  1. Приведите пример числа N, при вводе которого программа выдаёт неверный ответ. Укажите этот ответ. 2. Приведите пример числа N, при вводе которого программа выдаёт верный ответ. Укажите этот ответ. 3. Найдите все ошибки в этой программе (их может быть одна или несколько). Известно, что каждая ошибка затрагивает только одну строку и может быть исправлена без изменения других строк. Для каждой ошибки:  1) выпишите строку, в которой сделана ошибка; 2) укажите, как исправить ошибку, т.е. приведите правильный вариант строки. | | **25** | (№ 511) Дан массив, содержащий 2015 неотрицательных целых чисел. Необходимо найти в этом массиве количество таких элементов, которые равны среднему арифметическому двух элементов, расположенных непосредственно перед ним. Например, в массиве из 6 элементов, равных соответственно 4, 6, 5, 1, 3, 2, есть три таких элемента, они расположены на третьем, пятом и шестом месте и равны 5, 3 и 2.   |  |  |  | | --- | --- | --- | | Паскаль | Python | Си | | const n = 2015; var  a: array [0..n-1]   of integer;  i, k: integer; begin  for i:=0 to n-1 do  readln(a[i]);  ... end. | # допускается также # использовать  # целочисленную  # переменную k a = [] n = 2015 for i in range(n):  a.append(int(input())) ... | #include <stdio.h> #define n 2015 int main() {  int a[n];  int i, k;  for (i = 0; i < n; i++)  scanf("%d", &a[i]);  ...  return 0; } | | | **26** | (№ 530) Два игрока, Паша и Вася, играют в следующую игру. Перед игроками лежит куча камней. Игроки ходят по очереди, первый ход делает Паша. За один ход игрок может добавить в кучу **один или два камня** или увеличить количество камней в куче **в два раза**. Игра завершается в тот момент, когда количество камней в куче становится не менее **30**. Победителем считается игрок, сделавший последний ход, то есть первым получивший кучу, в которой будет 30 или больше камней. В начальный момент в куче было S камней, 1 ≤ S ≤29.   **Задание 1.** а) Укажите все такие значения числа S, при которых Паша может выиграть в один ход. Обоснуйте, что найдены все нужные значения S, и укажите выигрывающий ход для каждого указанного значения S. б)Укажите такое значение S, при котором Паша не может выиграть за один ход, но при любом ходе Паши Вася может выиграть своим первым ходом. Опишите выигрышную стратегию Васи.   **Задание 2.** Укажите 3 таких значения S, при которых у Паши есть выигрышная стратегия, причём Паша не может выиграть за один ход и может выиграть своим вторым ходом независимо от того, как будет ходить Вася. Для каждого указанного значения S опишите выигрышную стратегию Паши.   **Задание 3.** Укажите хотя бы одно значение S, при котором у Васи есть выигрышная стратегия, позволяющая ему выиграть первым или вторым ходом при любой игре Паши, и у Васи нет стратегии, которая позволит ему гарантированно выиграть первым ходом. Для указанного значения S опишите выигрышную стратегию Васи. Постройте дерево всех партий, возможных при этой выигрышной стратегии Васи (в виде рисунка или таблицы). | | **27** | (№ 555) На плоскости задано множество точек с целочисленными координатами. Необходимо найти минимально возможную площадь невырожденного (т.е. имеющего ненулевую площадь) треугольника, одна вершина которого расположена в начале координат, а две другие лежат на биссектрисах углов, образованных осями координат, и при этом принадлежат заданному множеству. Если такого треугольника не существует, необходимо вывести соответствующее сообщение. Напишите эффективную по времени и по используемой памяти программу для решения этой задачи. Программа считается эффективной по времени, если при увеличении количества точек в k раз время работы возрастает не более чем в k раз. Программа считается эффективной по памяти, если размер памяти для хранения всех необходимых данных не зависит от количества точек и не превышает 1 килобайта. Перед текстом программы кратко опишите алгоритм решения и укажите язык программирования и его версию.   В первой строке задаётся N – количество точек в заданном множестве. Каждая из следующих строк содержит два целых числа – координаты очередной точки. **Пример входных данных**:  3 6 6 -8 8 9 7  Если искомый треугольник существует, программа должна напечатать одно число: минимально возможную площадь треугольника, удовлетворяющего условиям. Если искомый треугольник не существует, программа должна напечатать сообщение: «Треугольник не существует». **Пример выходных данных** для приведённого выше примера входных данных:  48 | |