№ 24. Даны 4 натуральных числа. Необходимо выбрать из них и вывести на экран наибольшее нечётное число. Если нечётных чисел среди заданных нет, необходимо вывести слово «NO».

Для решения этой задачи ученик написал программу, но, к сожалению, его программа неправильная.

Ниже эта программа для Вашего удобства приведена на пяти языках программирования.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Бейсик** | |  | **Python** | | |
|  | |  |  | |  | |
|  | DIM M, X AS INTEGER M = 1  FOR I = 1 to 4 INPUT X  IF X MOD 2 >= 0 THEN IF X > M THEN  M = X END IF  END IF NEXT I  IF M > 0 THEN PRINT M  ELSE  PRINT "NO" END IF |  | m = 1  for i in range (4): x = int(input()) if x%2 >= 0:  if x > m: m = x  if m>0: print(m)  else: print("NO") |
|  | |
|  | |  |  | |  | |
|  | **Алгоритмический** **язык** |  | **Паскаль** |
|  | |  |  | |  | |
|  | алг нач  цел m, x, i m := 1  для i от 1 до 4 ввод x  если mod(x,2) >= 0 то если x > m то  m := x все  все кц  если m > 0 то вывод m  иначе  вывод "NO" все  кон |  | var m, x, i: integer; begin  m := 1;  for i:=1 to 4 do begin readln(x);  if x mod 2 >= 0 then begin if x > m then  m := x end  end;  if m > 0 then write(m)  else write('NO')  end. |
|  | |
|  | **C++** | | | | | |
|  | | | | | |  |
|  | #include <iostream> using namespace std; int main() {  int m, x, i; m = 1;  for (i=1; i<=4; ++i) { cin >> x;  if (x%2 >= 0) { if (x > m)  m = x; }  }  if (m > 0) cout << m;  else  cout << "NO"; return 0;  } | | | | |

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе чисел 3, 8, 5, 4.

2. Приведите пример исходных данных, содержащих хотя бы одно нечётное число, при вводе которых программа, несмотря на ошибки, выведет верный ответ.

3. Найдите в программе все ошибки (известно, что их не больше двух) и исправьте их. Для каждой ошибки выпишите строку, в которой она допущена, и приведите эту же строку в исправленном виде.

Достаточно указать ошибки и способ их исправления для одного языка программирования.

Обратите внимание: Вам нужно исправить приведённую программу, а не написать свою. Вы можете только заменять ошибочные строки, но не можете удалять строки или добавлять новые. Заменять следует только ошибочные строки: за исправления, внесённые в строки, не содержащие ошибок, баллы будут снижаться.

25. Дан массив, содержащий 2020 положительных целых чисел,

не превышающих 15 000. Необходимо найти минимальный элемент, значение которого не кратно 3, уменьшить все не кратные 3 элементы на величину этого минимума и вывести изменённый массив. Если в массиве нет элементов, не кратных 3, нужно вывести массив без изменений.

Например, для массива из 6 элементов, равных 2, 3, 5, 9, 12, 4, нужно получить и вывести массив, содержащий числа 0, 3, 3, 9, 12, 2.

Напишите на одном из языков программирования программу для решения этой задачи.

Исходные данные объявлены так, как показано ниже. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать часть из описанных.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Бейсик** | |  | **Python** | |
|  | |  |  | |  |
|  | CONST N=2020  DIM A(N) AS INTEGER DIM I, M, K AS INTEGER FOR I = 1 TO N  INPUT A(I) NEXT I  … END |  | # кроме уже указанных  # допускается использование # целочисленных переменных # m, k  a = []  N = 2020  for i in range(0, N): a.append(int(input()))  … |
|  | |
|  | | |  | | |
|  | **Алгоритмический** **язык** | |  | **Паскаль** | |
|  | |  |  | |  |
|  | алг нач  цел N=2020 целтаб a[1:N] цел i, m, k  нц для i от 1 до N ввод a[i]  кц …  Кон |  | const N=2020;  var  a: array [1..N] of integer; i, m, k: integer;  begin  for i:=1 to N do readln(a[i]);  … end. |
|  | | | | | |
|  | **C++** | | | | |
|  | | | | |  |
|  | #include <iostream> using namespace std; const int N=2020; int main(){  int a[N]; int i, m, k;  for (i=0; i<N; ++i) cin >> a[i];  …  return 0; } | | | |

В качестве ответа Вам необходимо привести фрагмент программы, который должен находиться на месте многоточия. Вы можете записать решение также на другом языке программирования (укажите название и используемую версию языка программирования, например, Free Pascal 2.6). В этом случае Вы должны использовать те же самые исходные данные и переменные, какие были предложены в условии.

26. Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежат две кучи камней. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может **добавить** в одну из куч **один** **камень** или **увеличить** количество камней **в** **куче** **в** **два** **раза**. Например, пусть в одной куче 6 камней, а в другой 8 камней; такую позицию мы будем обозначать (6, 8). За один ход из позиции (6, 8) можно получить любую из четырёх позиций: (7, 8), (12, 8), (6, 9), (6, 16). Чтобы делать ходы, укаждого игрока есть неограниченноеколичество камней. Игра завершается в тот момент, когда суммарное количество камней в кучах становится не менее 47. Победителем считается игрок, сделавший последний ход, то есть первым получивший позицию, в которой в кучах будет 47 или больше камней.

В начальный момент в первой куче было 11 камней, во второй куче – *S* камней, 1 ≤ *S* ≤ 35.

Будем говорить, что игрок имеет *выигрышную* *стратегию*, если он может выиграть при любых ходах противника. Описать стратегию игрока – значит описать, какой ход он должен сделать в любой ситуации, которая ему может встретиться при различной игре противника. В описание выигрышной стратегии не следует включать ходы играющего по ней игрока, которые не являются для него безусловно выигрышными, то есть не гарантируют выигрыш независимо от игры противника.

Выполните следующие задания.

**Задание** **1.**

а) Назовите все значения *S*, при которых Петя может выиграть первым ходом.

б) Петя сделал неудачный первый ход, после которого Ваня выиграл своим первым ходом. Назовитеминимальноезначение*S*, при котором это возможно.

**Задание** **2.**

Укажите такое значение *S*, при котором у Пети есть выигрышная стратегия, причём Петя не может выиграть первым ходом, но может выиграть своим вторым ходом независимо от того, как будет ходить Ваня. Для указанного значения *S* опишите выигрышную стратегию Пети.

**Задание** **3.**

Укажите такое значение *S*, при котором у Вани есть выигрышная стратегия, позволяющая ему выиграть первым или вторым ходом при любой игре Пети, и при этом у Вани нет стратегии, которая позволит ему гарантированно выиграть первым ходом.

Для указанного значения *S* опишите выигрышную стратегию Вани. Постройте дерево всех партий, возможных при этой выигрышной стратегии Вани (в виде рисунка или таблицы). В узлах дерева указывайте игровые позиции. Дерево не должно содержать партий, невозможных при реализации выигрывающим игроком своей выигрышной стратегии. Например, полное дерево игры не будет верным ответом на это задание.

1. На вход программы поступают *N* ≤1000 натуральных чисел, каждое из которых не превышает 10000. Необходимо определить количество пар элементов (*ai*, *aj*) этого набора, в которых 1 ≤ *i* < *j* ≤ *N*, сумма элементов нечётна, а произведение делится на 13. Напишите эффективную по времени и по памяти программу для решения этой задачи.

**Описание входных и выходных данных**

В первой строке входных данных задаётся количество чисел *N* (1 ≤ *N* ≤ 1000). В каждой из последующих *N* строк записано одно натуральное число, не превышающее 10000.

**Пример входных данных:**

**5**

**4**

**13**

**27**

**39**

**7**

**Пример выходных данных для приведённого выше примера входных данных:**

**2**

В приведённом наборе из 5 чисел имеются две пары (4, 13) и (4, 39), сумма элементов которых нечётна, и произведение кратно 13.

**Задача А.** Напишите программу для решения поставленной задачи, в которой входные данные будут запоминаться в массиве, после чего будут проверены все возможные пары элементов. Максимальная оценка за выполнение задания А – 2 балла.

**Задача Б.** Напишите программу для решения поставленной задачи, которая будет эффективна как по времени, так и по памяти. Максимальная оценка за выполнение задания Б – 4 балла.