```
{
  File:    equation.dpr
  Site:    www.acmu.ru

  For.     MSP

  Задача.  Кубическое уравнение

           (Время: 1 сек. Память: 16 Мб  Сложность: 56%)

           Написать программу, которая будет искать все целые X,
           удовлетворяющие уравнению
             A*X3 + B*X2 + C*X + D = 0, где A, B, C, D -
           заданные целые коэффициенты.

           Во входном файле INPUT.TXT записаны четыре целых
           числа: A, B, C, D. Все числа по модулю не превышают
           2*10^9.

           В выходной файл OUTPUT.TXT выведите сначала количество
           решений этого уравнения в целых числах, а затем сами
           корни в возрастающем порядке.
           Если уравнение имеет бесконечно много корней, то следует
           вывести в выходной файл одно число -1 (минус один).

           Примеры

           N    input.txt     output.txt
           -----------------------------
           1    1 0 0 -27     1 3
           -----------------------------
           2    0 1 2 3       0
           -----------------------------

  Решение. Калмыков Вадим (ProCrypt),
           г. Сургут, ЦНИТ "Северная Звезда",
           02:45, 23.03.2008

  Source : http://acm.dvpion.ru

  Comment.
    Первое, что хотелось бы сказать, это то, что данный алгоритм при
    его правильной модификации позволяет решать уравнения высших
    степеней без использования длинной арифметики. Решение проблемы
    больших чисел основано на теории алгебры многочленов, а именно
    на делимости коэффициентов на корни уравнения. Чтобы не
    производить громоздкие вычисления, используется один
    из основных методов решения алгебраических уравнений - замена
    переменной.
```

```
    В этой программе данный метод применен немного по - другому, а
    именно, заменяется константа переменной до тех пор, пока не
    получится линейное уравнение, тем самым, например, для решения
    уравнения степени 100 нам потребуется всего лишь 99 элементов
    массива, после чего будет получено элементарное линейное
    уравнение.
}

{$R-}

{
  Директива $R- вовсе необязательна, программа будет работать на
  локальной машине и без нее, но в программе она имеет место, так
  как решение задачи проверялось на сайте с системой автоматического
  тестирования acm.dvpion.ru

  При отсутствии корней уравнения, при сортировки массива корней
  получается следующее выражение
    for i := 1 to -1 do,
  цикл не выполняется ни разу, но система проверки генерирует
  ошибку, дабы избежать проблем с зависанием системы.
}

Program Equation;

Var
  i : LongInt; // счетчик циклов

  // R [0] - число корней, R [1]..R[3] - корни
  R : array [0..3] of LongInt;

  x : LongInt; // предполагаемый корень

  n : LongInt; // свободный член, раскладываемый на множители

  Coef : Array [1..4] of LongInt; // коэффициенты уравнения

// функция проверки принадлежности числа ко множеству корней
Function F (x : LongInt) : Boolean;

var
  q,
  y : LongInt; // замены констант, их значение подробно описано ниже

begin
  {
    Если X является корнем уравнения, то он является делителем
    свободного члена.

    Следовательно, разделив уравнение ax^3 + bx^2 + cx + d = 0 на x,
    который по предположению является корнем уравнения и не равен
```

```
    нулю, то получим равносильное выражение (но никак не уравнение!)
      ax^2 + bx + c + d/x = 0,
    причем d/x является целым числом, так как x делитель d. Отсюда
    вытекает, что
      c + d/x тоже целое число, так как по условию коэффициенты -
    целые числа.

    Заменим выражение c + d/x = q. Тогда имеет место равенство
    ax^2 + bx + q = 0, так как мы получили равносильное выражение,
    отсюда следует, что данное выражение тоже кратно x, а именно,
    нас интересует коэффициент q, так называемый свободный член
    уравнения.

    Повторное деление выражения на x дает следующий результат:
      ax + b + q/x = 0.

    Аналогично получается, что b + q/x целое число, заменим его
    через y и получим ax + y = 0.

    Теперь нам необходимо проверить истинность условия, и если
    условие истинно, значит x является корнем уравнения.

    При этом x = -y / a.
    А значит, -y / a целое число (по условию корни - целые числа).

    Следует учесть еще один случай, когда a = 0 (Coef [1] = 0).
    В этом случае может происходить деление на ноль, но чтобы
    избежать ошибки времени исполнения мы учтем следующую
    возможность:

    Если a = 0, то имеем 0x + y = 0, следовательно y = 0
  }

  // свободный член должен быть кратен корню уравнения
  if (Coef [4] mod x <> 0)
    then
      begin
        F := False;

        Exit;
      end;

  q := Coef [3] + Coef [4] div x; // вычисляем первую замену

  // эта замена должна быть кратна x
  if (q mod x <> 0)
    then
      begin
        F := False;

        Exit;
```

```
      end;
  y := Coef [2] + q div x; // вычисляем вторую замену
  if (Coef [1] = 0)
    then
     if (y = 0)
       then
         begin
           F := True;
           Exit;
         end
       else
         begin
           F := False;
           Exit;
         end;
  // проверяем условие ax + y = 0
  if (y mod Coef [1] = 0)
     and
     (- y div Coef [1] = x)
    then
      F := True
    else
      F := False;
end;

// функция, проверяющая, не повторяется ли корень
Function Rep (x : Integer) : Boolean;
var
  i : byte;
begin
  for i := 1 to 3 do
    if (R [i] = x)
      then
        begin
          Rep := True;
          Exit;
        end;
  Rep := False;
end;
```

```
Begin
  Assign (Input, 'input.txt');
  Assign (Output, 'output.txt');

  Reset (Input);
  Read (Coef [1], Coef [2], Coef [3], Coef [4]);
  Close (Input);

  {
    Ищем первый ненулевой коэффициент в порядке возрастания степени
    аргумента
  }
  n := 0;
  for i := 4 downto 1 do
    if (Coef [i] <> 0)
      then
        begin
          n := Coef [i];

          Break;
        end;

  // если ничего не найдено - все коэффициенты нулевые, значит
  // тождество
  if (n = 0)
    then
      begin
        ReWrite (Output);
        Write (-1);
        Close (Output);

        Exit;
      end;

  // производим инициализацию массива
  for i := 0 to 3 do
    R [i] := 0;

  {
    Если число число X является действительным корнем уравнения,
      то это число является делителем свободного члена уравнения
  }

  // если свободный член равен нулю, то 0 - корень
  if (Coef [4] = 0)
    then

      // массив инициализирован нулями, достаточно просто
      // увеличить число корней
      Inc (R [0]);
  {
```

```
  Все простые делители числа n располагаются на промежутке
  [1; Sqrt (n)]
  Поэтому все остальные делители могут быть получены
  делением n на x

  Но помимо натуральных делителей числа следует еще учитывать
  числа противоположные делителяем и число 0.
}

n := Abs (n); // невозможно извлечь корень из отрицательного числа
for x := 1 to Trunc (Sqrt (n)) do
  if (n mod x = 0)
    then
      begin
        { Проверяем x }
        if (F (x))
           and
           not (Rep (x))
          then
            begin
              Inc (R [0]);
              R [R [0]] := x;
            end;

        { Теперь число, противоположное x }
        if (F (-x))
           and
           not (Rep (-x))
          then
            begin
              Inc (R [0]);
              R [R [0]] := -x;
            end;

        { Частное от деления n на x тоже может быть корнем }
        if (F (n div x))
           and
           not (Rep (n div x))
          then
            begin
              Inc (R [0]);
              R [R [0]] := n div x;
            end;

        { Аналогично и ему противоположное }
        if (F (- n div x))
           and
           not (Rep (- n div x))
          then
            begin
              Inc (R [0]);
```

```
                R [R [0]] := - n div x;
              end;
        end;

  // сортируем корни в порядке возрастания
  for i := 1 to (R [0] - 1) do
    for n := 1 to (R [0] - i) do
      if (R [n] > R [n + 1])
        then
          begin
            x := R [n];
            R [n] := R [n + 1];
            R [n + 1] := x;
          end;

  ReWrite (Output);

  Write (R [0]);

  for i := 1 to R [0] do
    Write (' ', R [i]);

  Close (Output);

  {
    В завершении хотелось бы добавить, что нет ничего невозможного,
    если сильно захотеть!
    Это относится к предупреждению авторов задачи и сайта,
    что данная задача решается только с помощью Delphi и современных
    компиляторов языка C, поддерживающих такие типы данных, как
    Int64 или long long (язык C++),
    или с помощью встроенной длинной арифметики Java, но никак не на
    Паскале.

    Оригинальное решение на Паскале было отправлено на проверку и
    прошло все тесты в течение 0.01 сек, в то время как аналог
    авторов сайта прошел за 0.47 сек.

    И пусть это решение займет должное место на сайте
    http://www.Best-Listing.ru моего прекрасного учителя
    Сергея Петровича Митрофанова, побудившего меня заняться
    программированием, вложившего в меня свои силы, огромный труд и
    терпение на протяжении долгого времени, за что ему
    ОГРОМНАЯ БЛАГОДАРНОСТЬ.

    Калмыков В.В.
    23 марта 2008 года
  }
End.
```

Listing данной задачи опубликован в сети Internet по адресу http://www.Best-Listing.ru/color-10-task-647.html

Sergey Mitrofanov, 11.09.14, 12:03
E-mail: infostar@mail.ru