



**ЕГЭ 2021. ИНФОРМАТИКА
ПРОГРАММИРОВАНИЕ**

**Как решать
задачи
ExamBegin
1-80?**

12+

Елена Леонидовна Теплоухова

ЕГЭ 2021. Информатика.

Программирование. Как

решать задачи ExamBegin 1-80?

http://www.litres.ru/pages/biblio_book/?art=63577897

SelfPub; 2023

Аннотация

Если вы планируете сдавать ЕГЭ по информатике, то наверняка задачи на программирование представляют для вас самую большую проблему. В этой книге доступно даётся пошаговый алгоритм решения 80 первых задач из ExamBegin. С этим сборником задачи на программирование больше не будут вас пугать, и вы сможете получить максимальный балл на экзамене.

Содержание

Введение	5
Условные операторы и циклы	8
ExamBegin1	8
ExamBegin2	9
ExamBegin3	10
ExamBegin4	11
ExamBegin5	13
ExamBegin6	14
ExamBegin7	15
ExamBegin8	16
ExamBegin9	17
ExamBegin10	18
ExamBegin11	19
ExamBegin12	20
ExamBegin13	21
ExamBegin14	22
ExamBegin15	23
ExamBegin16	24
ExamBegin18	26
ExamBegin19	28
ExamBegin20	30
Формирование массивов	32
ExamBegin21	32

Елена Теплоухова

ЕГЭ 2021. Информатика.

Программирование.

Как решать задачи

ExamBegin 1-80?

Введение

В ЕГЭ по предмету информатика одной из главных тем является программирование. Тема сложная для самостоятельного изучения и требующая понимания ее сути. В программе Паскаль.АВС для тренировки представлены сотни задач различных уровней сложности. Вроде бы столько возможностей хорошо подготовиться к ЕГЭ, много систематизированных заданий – берите и решайте. Но, прочитывая задачи, вы понимаете, что не знаете как их решить. На занятиях с репетитором вы разбирали подобные задачи, но немного измененные условия в задаче вводят в ступор. И появляется страх, что на экзамене попадется именно эта формулировка задачи.

Вы начинаете поиск подобных задач на программирова-

ние, которыми пестрят книги и сайты по подготовке к ЕГЭ, но объяснения представлены сложными способами, мало пояснений к решениям, не все нюансы разных условий задач отражены. После поисков в интернете, зачастую безрезультатных, и блужданий по сомнительным форумам, чаще всего приходится ждать следующей консультации с репетитором. Все, что вам нужно в этой ситуации для экономии времени и денег – это иметь простые и понятные решения, чтобы решать любые формулировки заданий и понимать суть их решения. Это дает уверенность в своих силах, систематизирует знания и формирует опыт решения таких задач.

Работая учителем информатики, я столкнулась с необходимостью систематизировать свои знания по теме программирование. Многие задачи оказались для меня хорошим поводом углубиться в эту тему, найти более простые и понятные способы решения. Моей целью было не только прорешать все задания, но сгруппировать их – такая систематизация позволит намного быстрее усвоить алгоритмы решения и безошибочно определить – какой путь решения выбрать для той задачи, с которой вы столкнулись. Результатом систематизации стал сборник задач, выстроенный от самых простых задач до задач повышенной сложности. В нем представлены решения 80 задач, что собраны для тренировки к заданиям ЕГЭ по теме программирование.

Сборник позволит вам с минимальными временными и финансовыми затратами на репетитора самим освоить тему.

Он выстроен таким образом, что вы ощутите себя на индивидуальном занятии и сможете продвигаться в своем темпе.

Условные операторы и циклы

ExamBegin1

На вход подаются два целых числа; числа расположены в одной строке. Вывести вначале максимальное, а затем минимальное из них. Числа должны выводиться в одной строке и разделяться одним пробелом.

```
uses PT4Exam;  
var  
a, b: integer;  
begin  
Task('ExamBegin1');  
Read(a, b);  
if a > b then Write(a, ' ', b)  
else Write(b, ' ', a);  
end.
```


ExamBegin2

На вход подаются три вещественных числа; числа расположены в одной строке. Вывести вначале минимальное, а затем максимальное из них. Каждое число должно выводиться на новой строке и снабжаться комментарием: «MIN=» для минимального, «MAX=» для максимального.

```
uses PT4Exam;
var
  x, max, min: real;
  i: integer;
begin
  Task('ExamBegin2');
  min := maxreal;
  max := -maxreal;
  for i := 1 to 3 do
    begin
      read(x);
      if x > max then max := x;
      if x < min then min := x;
    end;
    WriteLn('MIN=', min:4:2);
    WriteLn('MAX=', max:4:2);
  end.
```

ExamBegin3

На вход подаются четыре целых числа; каждое число вводится с новой строки. Вывести минимальное из них.

```
uses PT4Exam;  
var  
i, a, min: integer;  
begin  
Task('ExamBegin3');  
min := 32768;  
for i := 1 to 4 do  
begin  
Read(a);  
if a < min then min := a;  
end;  
Write(min);  
end.
```

ExamBegin4

На вход подаются три целых числа – коэффициенты A , B , C квадратного уравнения $A \cdot x^2 + B \cdot x + C = 0$ (A не равно 0). Каждое число вводится с новой строки. Найти корни данного уравнения. Корни вычисляются как вещественные числа. Если имеются два различных корня, то вначале выводить меньший, а затем больший (каждый на новой строке). Если корней нет, то вывести текст «NO».

```
uses PT4Exam;  
var  
a, b, c: integer;  
d, x, x1, x2: real;  
begin  
Task('ExamBegin4');  
Read(a, b, c);  
d := sqr(b) - 4 * a * c;  
if d < 0 then Write('NO')  
else begin  
if d = 0 then begin  
x := -(b / (2 * a));  
WriteLn(x:4:2);  
end  
else begin
```

```
x1 := -(b + sqrt(d)) / (2 * a);  
x2 := -(b - sqrt(d)) / (2 * a);  
if x1 > x2 then begin  
  WriteLn(x2:4:2);  
  WriteLn(x1:4:2);  
end  
else begin  
  WriteLn(x1:4:2);  
  WriteLn(x2:4:2);  
end;  
end;  
end;  
end.
```

ExamBegin5

На вход подаются два целых положительных числа; каждое число вводится с новой строки. Найти их наибольший общий делитель (НОД), используя алгоритм Евклида: $\text{НОД}(A, B) = \text{НОД}(B, A \bmod B)$, если $B \neq 0$; $\text{НОД}(A, 0) = A$, где «mod» обозначает операцию взятия остатка от деления.

```
uses PT4Exam;  
var  
a, b: integer;  
begin  
  Task('ExamBegin5');  
  Read(a, b);  
  while a <> b do  
    begin  
      if a > b then a := a - b  
      else b := b - a;  
    end;  
  Write(a);  
end.
```

ExamBegin6

На вход подаются два целых положительных числа; числа расположены в одной строке. Найти их наименьшее общее кратное (НОК), используя формулу $\text{НОК}(A, B) = A \cdot (B / \text{НОД}(A, B))$, где $\text{НОД}(A, B)$ – наибольший общий делитель, а символ «/» обозначает операцию деления нацело. НОД находить с помощью алгоритма Евклида (см. задание ExamBegin5).

```
uses PT4Exam;  
var  
a, b, nod: integer;  
begin  
Task('ExamBegin6');  
Read(a, b);  
nod := a * b;  
while a <> b do  
begin  
if a > b then a := a - b  
else b := b - a;  
end;  
Write(nod div a);  
end.
```

ExamBegin7

На вход подаются два целых положительных числа: A и D ($2 \leq D \leq 9$); числа расположены в одной строке. Вывести строку, изображающую запись числа A в системе счисления с основанием D (полученная строка не должна содержать незначащих нулей).

```
uses PT4Exam;  
var  
a, b, rem, num, k: integer;  
begin  
Task('ExamBegin7');  
Read(a, b);  
num := 0; k := 1;  
while a > 0 do  
begin  
rem := a mod b;  
a := a div b;  
num := num + rem * k;  
k := k * 10;  
end;  
Write(num);  
end.
```

ExamBegin8

На вход в первой строке подается целое число D ($2 \leq D \leq 9$), а во второй строке – текст, изображающий запись целого положительного числа A в системе счисления с основанием D . Вывести число A в десятичной системе счисления.

```
uses PT4Exam;  
var  
a, num, k, m: integer;  
b: string;  
begin  
Task('ExamBegin8');  
Readln(a);  
Readln(b);  
num := 0; m := 1;  
while length(b) > 0 do  
begin  
k := StrToInt(Copy(b, length(b), 1));  
Delete(b, length(b), 1);  
num := num + k * m;  
m := m * a;  
end;  
Write(num);  
end.
```


ExamBegin9

На вход подается целое положительное число. Если оно является степенью числа 2, то вывести текст «YES», иначе вывести «NO».

```
uses PT4Exam;  
var  
a, x: integer;  
num: boolean;  
begin  
Task('ExamBegin9');  
ReadLn(a);  
x := 1; num := false;  
while x <= a do  
begin  
x := x * 2;  
if (x = a) or (a = 1) then num := true;  
end;  
if num = true then Write('YES')  
else Write('NO');  
end.
```

ExamBegin10

На вход подаются два целых положительных числа A и K ; каждое число располагается на новой строке. Если число A является степенью числа K , т. е. $A = K^N$, то вывести показатель этой степени N , иначе вывести текст «NO».

```
uses PT4Exam;
var
  a, k, x, n: integer;
  z: boolean;
begin
  Task('ExamBegin10');
  Read(a, k);
  x := 1; n := 0; z := false;
  while x < a do
    begin
      x := x * k;
      n := n + 1;
      if x = a then z := true;
    end;
  if (z = true) or (a = 1) then write(n)
  else write('NO');
end.
```

ExamBegin11

На вход подается целое число, большее 1. Вывести наименьший делитель данного числа, больший 1.

```
uses PT4Exam;
```

```
var
```

```
a, x: integer;
```

```
begin
```

```
Task('ExamBegin11');
```

```
Read(a);
```

```
x := 2;
```

```
while a mod x <> 0 do
```

```
x := x + 1;
```

```
Write(x);
```

```
end.
```

ExamBegin12

На вход подается целое число, большее 1. Если оно является простым, т. е. не имеет положительных делителей, кроме 1 и самого себя, то вывести текст «YES», иначе вывести «NO».

```
uses PT4Exam;  
var  
a, i, k: integer;  
begin  
  Task('ExamBegin12');  
  Read(a);  
  k := 0;  
  for i := 1 to a do  
    if a mod i = 0 then k := k + 1;  
    if k > 2 then Write('NO')  
    else Write('YES');  
  end.
```

ExamBegin13

На вход подается целое число, большее 1. Разложить данное число на простые множители и вывести все множители в порядке возрастания (каждое число выводить на новой строке, среди выводимых чисел могут быть одинаковые).

```
uses PT4Exam;  
var  
a, k: integer;  
begin  
Task('ExamBegin13');  
Read(a);  
k := 2;  
while a <> 1 do  
begin  
if a mod k = 0 then begin  
Writeln(k);  
a := a div k;  
end  
else k := k + 1;  
end;  
end.
```

ExamBegin14

На вход в первой строке подается целое положительное число N, а во второй строке – набор из N целых чисел. Вывести сумму всех чисел из исходного набора.

```
uses PT4Exam;  
var  
a, n, i, sum: integer;  
begin  
Task('ExamBegin14');  
Read(n);  
sum := 0;  
for i := 1 to n do  
begin  
read(a);  
sum := sum + a;  
end;  
Write(sum);  
end.
```

ExamBegin15

На вход в первой строке подается целое положительное число N , а в следующих N строках – набор из N вещественных чисел (по одному числу в строке). Вывести произведение всех чисел из исходного набора.

```
uses PT4Exam;  
var  
x, i: integer;  
a, p: real;  
begin  
  Task('ExamBegin15');  
  Read(x);  
  p := 1;  
  for i := 1 to x do  
    begin  
      read(a);  
      p := p * a;  
    end;  
  Write(p:4:2);  
end.
```

ExamBegin16

На вход подается набор целых чисел, содержащий одно или более ненулевых чисел и завершающийся числом 0; каждое число располагается на новой строке. Вывести сумму всех чисел из исходного набора .

```
uses PT4Exam;  
var  
a, sum: integer;  
begin  
Task('ExamBegin16');  
sum := 0;  
ReadLn(a);  
while a <> 0 do  
begin  
sum := sum + a;  
ReadLn(a);  
end;  
WriteLn(sum);  
end.
```

ExamBegin17

На вход подается набор вещественных чисел, содержащий одно или более положительных чисел и завершающийся отрицательным числом; каждое число

располагается на новой строке. Вывести произведение всех положительных чисел из исходного набора.

```
uses PT4Exam;  
var  
a, sum: real;  
begin  
  Task('ExamBegin17');  
  sum := 1;  
  ReadLn(a);  
  while a > 0 do  
    begin  
      sum := sum * a;  
      readln(a);  
    end;  
  WriteLn(sum:4:2);  
end.
```

ExamBegin18

На вход подается целое положительное число N . Сформировать и вывести целочисленный массив размера N , содержащий N первых положительных нечетных чисел: 1, 3, 5, Каждое число выводить на новой строке.

```
uses PT4Exam;
const
  R = 100;
var
  m: array[1..R] of integer;
  a, k, n: integer;
begin
  Task('ExamBegin18');
  Read(a);
  k := 1; n := 0;
  while n < a do
    begin
      if k mod 2 = 1 then begin
        m[k] := k;
        n := n + 1;
        Writeln(m[k]);
      end;
      k := k + 1;
```

```
end;  
end.
```

ExamBegin19

На вход подается целое положительное число N . Сформировать и вывести целочисленный массив размера N , содержащий степени двойки от первой до N -й: 2, 4, 8, 16, Выводить все числа в одной строке, для каждого числа отводить 5 экранных позиций.

```
uses PT4Exam;
const
  R = 100;
var
  m: array[0..R] of integer;
  a, i: integer;
begin
  Task('ExamBegin19');
  Read(a);
  m[0] := 1;
  for i := 1 to a do
  begin
    m[i] := m[i - 1] * 2;
    Write(m[i]:5);
  end;
end.
```

Другое решение 19 задачи.

```
uses PT4Exam;
```

```
var
  a, i, p: integer;
begin
  Task('ExamBegin19');
  Read(a);
  p := 1;
  for i := 1 to a do
    begin
      p := p * 2;
      Write(p:5);
    end;
  end.
```

ExamBegin20

На вход подаются три числа: целое число N (> 1) и вещественные числа A и D – соответственно первый член и разность арифметической прогрессии (каждое число вводится с новой строки). Сформировать и вывести массив размера N , содержащий N первых членов данной прогрессии: $A, A + D, A + 2 \cdot D, A + 3 \cdot D, \dots$. Все числа выводить в одной строке, для каждого числа отводить 7 экранных позиций.

```
uses PT4Exam;  
const  
  R = 100;  
var  
  m: array[1..R] of real;  
  n, i: integer;  
  a, d: real;  
begin  
  Task('ExamBegin20');  
  Read(n, a, d);  
  for i := 1 to n do  
    begin  
      m[i] := a;  
      Write(m[i]:7:2);  
      a := a + d;
```

```
end;
```

```
end.
```

Другое решение 20 задачи.

```
uses PT4Exam;
```

```
var
```

```
  n, i: integer;
```

```
  a, d: real;
```

```
begin
```

```
  Task('ExamBegin20');
```

```
  Read(n, a, d);
```

```
  for i := 1 to n do
```

```
    begin
```

```
      Write(a:7:2);
```

```
      a := a + d;
```

```
    end;
```

```
end.
```

Формирование массивов

ExamBegin21

На вход подаются три числа: целое число $N (> 1)$ и вещественные числа A и D – соответственно первый член и знаменатель *геометрической прогрессии* (каждое число вводится с новой строки). Сформировать и вывести массив размера N , содержащий N первых членов данной прогрессии: $A, A \cdot D, A \cdot D^2, A \cdot D^3, \dots$. Каждое число выводить на новой строке.

```
uses PT4Exam;  
const R = 100;  
var  
m: array[1..R] of real;  
n, i: integer;  
a, d: real;  
begin  
Task('ExamBegin21');
```


Конец ознакомительного фрагмента.

Текст предоставлен ООО «ЛитРес».

Прочитайте эту книгу целиком, [купив полную легальную версию](#) на ЛитРес.

Безопасно оплатить книгу можно банковской картой Visa, MasterCard, Maestro, со счета мобильного телефона, с платежного терминала, в салоне МТС или Связной, через PayPal, WebMoney, Яндекс.Деньги, QIWI Кошелек, бонусными картами или другим удобным Вам способом.