Единый государственный экзамен по ИНФОРМАТИКЕ и ИБТ

Инструкция по выполнению работы

Экзаменационная работа состоит из двух частей, включающих в себя

27 заданий. Часть 1 содержит 23 задания с кратким ответом, часть 2 содержит 4 задания с развёрнутым ответом.

На выполнение экзаменационной работы по информатике и ИКТ отводится 3 часа 55 минут (235 минут).

Ответы к заданиям 1—23 записываются в виде числа, последовательности букв или цифр. Ответы запишите в поле ответа в тексте работы, а затем перенесите в бланк ответов № 1.



Задания 24—27 требуют развёрнутого ответа. В бланке ответов № 2 укажите номер задания и запишите его полное решение.

Все бланки ЕГЭ заполняются яркими чёрными чернилами. Допускается использование гелевой, капиллярной или перьевой ручек.

При выполнении заданий можно пользоваться черновиком. Записи в черновике не учитываются при оценивании работы.

Баллы, полученные Вами за выполненные задания, суммируются. Постарайтесь выполнить как можно больше заданий и набрать наибольшее количество баллов.

*Желаем успеха.!*

В экзаменационных заданиях используются следующие соглашения.

1. Обозначения для логических связок (операций):

а) отрицание (инверсия, логическое НЕ) обозначается (например, =А);

b) конъюнкция (логическое умножение, логическое И) обозначается /\ (например, А /\ В) либо & (например, А & В);

 дизъюнкция (логическое сложение, логическое ИЛИ) обозначается \/ (например, А \/ В) либо Ј(например, А ЈВ);

d) следование (импликация) обозначается ---г (например, А --—г В);

е) тождество обозначается - (например, А - В). Выражение А - В истинно тогда и только тогда, когда значения А и В совпадают (либо они оба истинны, либо они оба ложны);

 символ 1 используется для обозначения истины (истинного высказывания); символ 0 — для обозначения лжи (ложного высказывания).

1. Два логических выражения, содержащих переменные, называются равносильными (эквивалентными), если значения этих выражений совпадают при любых значениях переменных. Так, выражения А ———+ В и (=А) \/ В равносильны, а А \/ В и А /\ В неравносильны (значения выражений разные, например, при А = 1, В — 0).
2. Приоритеты логических операций: инверсия (отрицание), конъюнкция (логическое умножение), дизъюнкция (логическое сложение), импликация (следование), тождество. Таким образом, =А /\ В \/ С /\ D означает то же, что и ((=А) /\ В) \/ (С /\ D).

Возможна запись А /\ В /\ С вместо (А /\ В) /\ С. То же относится и к дизъюнкции: возможна запись А \/ В \/ С вместо (А \/ В) \/ С.

1. Обозначения Мбайт и Кбайт используются в традиционном для информатики смысле — как обозначения единиц измерения, чьё соотношение с единицей «байт» выражается степенью двойки.

Часть 1

*Ответами к заданиям 1—23 являются число, последовательность букв или цифр, которые следует записать в БЛАНК OTBETOB Х• 1 справа от номера соответствующего задания, начиная с первой илеточки, без nPoбeлoв, запятъіх и других допоянитеяьных символов. Каждъій символ nuшиme в отдельной клеточве в соответствии с приведёнными в dланке образцами.*

 Сколько существует целых чисел, удовлетворяющих неравенству 110010112 х CF 6?

В ответ укажите только количество таких чисел, сами числа указывать не нужно.

Ответ:

2 Логическая функция *Е* задаётся выражением

*=х z) =w) (=z =w).*

На рисунке приведён фрагмент таблицы истинности функции *Е,* содержащий все наборы аргументов, при которых функция *Г-* ложна. Определите, какому столбцу таблицы истинности функции *Е* соответствует каждая из переменных х, у, *z, w.*

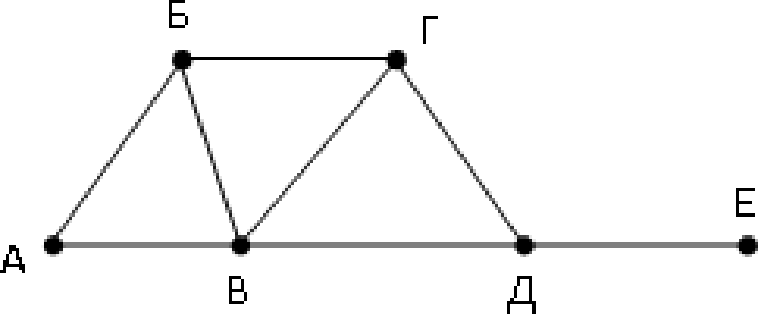
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  | F |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 |

В ответе напишите буквы х, у, z, w в том порядке, в котором идут соответствующие им столбцы.

Ответ:

ј На рисунке справа схема дорог Н-ского района изображена в виде графа, в таблице содержатся сведения о длинах этих дорог (в километрах).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | П1 | П2 | ПЗ | П4 | П5 | Пб |
| П1 |  | 10 |  |  | 8 | 5 |
| П2 | 10 |  |  | 20 | 12 |  |
|  |  |  |  | 4 |  |  |
| П4 |  | 20 | 4 |  | 15 |  |
| П5 | 8 | 12 |  | 15 |  | 17 |
| П6 | 5 |  |  |  | 17 |  |



Так как таблицу и схему рисовали независимо друг от друга, то нумерация населённых пунктов в таблице никак не связана с буквенными обозначениями на графе. Определите длину кратчайшего маршрута между пунктами А и В. Передвигаться можно только по указанным дорогам.

Ответ:

 В фрагменте базы данных представлены сведения о родственных отношениях. На основании приведенных данных определите, сколько детей

родилось, когда их отцам было менее 28 лет? Таблица 1

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Фамилия И.О. | Пол | Год рождепия |
| 1243 | Бесчастных П.А. | М | 1993 |
| 1248 | Попович А.А. | М | 1999 |
| 1250 | Ан Н.А. | Ж | 1994 |
| 1251 | Ан В.А. | Ж | 1997 |
| 1257 | Фоменко П.И. | М | 2001 |
| 2230 | Фоменко Е.А. | Ж | 1972 |
| 2300 | Фоменко И.А. | М | 1976 |
| 3252 | Фоменко Т.Х. | Ж | 1974 |
| 3293 | Поркуян А. А | Ж | 1997 |
| 3319 | Сабо С.А. | Ж | 1995 |
| 5215 | Фоменко А.К. | М | 1947 |
| 6214 | Попович Л.П. | Ж | 1942 |
| 6258 | Фоменко Т.И. | Ж | 1997 |
| 9252 | Бесчастных А.П | М | 1966 |

Таблица 2

|  |  |
| --- | --- |
| ID Родителя | ID Ребенка |
| 2230 | 1243 |
| 2230 | 1251 |
| 2230 | 3319 |
| 2300 | 6258 |
| 2300 | 1257 |
| 3252 | 6258 |
| 3252 | 1257 |
| 5215 | 2230 |
| 5215 | 2300 |
| 6214 | 2230 |
| 6214 | 2300 |
| 9252 | 1243 |
| 9252 | 1251 |
| 9252 | 3319 |

 По каналу связи с помощью равномерного двоичного кода передаются сообщения, содержащие только 4 буквы А, Б, В, F. Каждой букве

соответствует своё кодовое слово, при этом для набора кодовых слов выполнено такое свойство: любые два слова из набора отличаются не менее чем в трёх позициях. Это свойство важно для расшифровки сообщений при наличии помех. Для кодирования букв Б, В, Г используются 5-битовые кодовые слова: Б: 00001, В: 01111, Г: 10110. 5-битовый код для буквы А начинается с 1 и заканчивается на 0. Определите кодовое слово для буквы А.

Ответ:

 На вход алгоритма подаётся натуральное число N. Алгоритм строит по нему новое число R следующим образом.

1. Строится двоичная запись числа N.
2. К этой записи дописывается справа бит чётности: 0, если в двоичном коде числа N было чётное число единиц, и 1, если нечётное.
3. К полученному результату дописывается ещё один бит чётности.

Полученная таким образом запись (в ней на два разряда больше, чем в записи исходного числа N) является двоичной записью искомого числа R. Укажите минимальное число N, после обработки которого с помощью этого алгоритма получается число, большее, чем 184. В ответе это число запишите в десятичной системе.

Ответ:

 Дан фрагмент электронной таблицы. Из ячейки E4 в ячейку D3 была скопирована формула. При копировании адреса ячеек в формуле

автоматически изменились. Каким стало числовое значение ячейки D3:

Ответ:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | А | В | С | D | Е |
| 1 | 40 | 4 | 400 | 70 | 7 |
| 2 | 30 | 3 | 300 | 60 | 6 |
| 3 | 20 | 2 | 200 |  | 5 |
| 4 | 10 | 1 | 100 | 40 | = $B2 \* С$З |

Ответ:

 Запишите число, которое будет напечатано в результате выполнения следующей программы.



 Все 5-буквенные слова, составленные из букв Л, Е, М, У,Р, записаны в алфавитном порядке и пронумерованы, начиная с 1. Ниже приведено

начало списка.

* 1. EEEEE
  2. ЕЕЕЕЛ
  3. EEEEM
  4. EEEEP
  5. EEEEУ
  6. ЕЕЕЛЕ

Укажите номер первого слова, которое начинается с буквы Р.

Ответ:

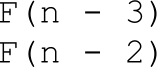
Ниже на пяти языках программирования записан рекурсивный алгоритм F.

|  |  |
| --- | --- |
| Бейсик | Pphon |
| DIM S, К AS INTEGER | s = 0  k = 0  while k < 12:  s = s + 2\*k k = k + 3  print(s) |
| S = 0 |
| К = 0 |
| WHILE К < 12 |
| S = S + 2\*к |
| WEND |
| PRINT S |
| Алгоритмический язык | Паскаль |
| али | var k, s: integer;  begin  s:=0;  k:=0;  while k < 12 do begin s:=s+2\*k;  end; write(s);  end. |
| нач |
| цел п, s |
| s := 0 |
| п := 0 |
| нц пока k 12 |
| s := s+2\*k |
| С++ | |
| #include <iostream>  using namespace std;  int main() (  int s = 0, k = 0; while (k < 12) (  s = s+2\*k;; k k+3;  cout << s << endl; return 0; | |









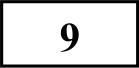






Ответ:

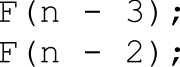
|  |  |
| --- | --- |
| **Бейсик** | Python |
| SUв F(n)  IF п >= 3 THEN PRINT п  Г(п - 3)  Г(п - 2) ГND IF  ГND SUB | def F(n):  if п >= 3: print(n) |
| **Алгоритмический язык** | **Маскаль** |
| алг F(цел п)  нач  если п >= 3 то вывод п  Г(п - 3)  F(n - 2)  все  кОн | procedure F(n: integer);  begin  if п >= 3 then begin write(n);  Г(п - 3);  F(n - 2)  end end; |
|  | |
| void Г(int п)(  if (п >= 3)( std::cout <<п; | |

 Рисунок размером 512 на 128 пикселей занимает в памяти 32 Кбайт.

Найдите максимально возможное количество цветов в палитре

изображения.

Ответ:





Что выведет программа при вызове F(7)? Числа должны быть записаны в том же порядке, в котором они выводятся на экран. Ответ:

В терминологии сетей TCP/IP маска сети – это двоичное число, меньшее 232; в маске сначала (в старших разрядах) стоят единицы, а затем с некоторого места нули. Маска определяет, какая часть IP-адреса узла сети относится к адресу сети, а какая – к адресу самого узла в этой сети. Обычно маска записывается по тем же правилам, что и IP-адрес – в виде четырёх байт, причём каждый байт записывается в виде десятичного числа. Адрес сети получается в результате применения поразрядной конъюнкции к заданному IP-адресу узла и маске.

**12**

Например, если IP-адрес узла равен 131.32.255.131, а маска равна 255.255.240.0, то адрес сети равен 131.32.240.0.

Для узла с IP-адресом 163.232.136.60 адрес сети равен 163.232.136.0. Найдите наибольшее возможное количество единиц в двоичной записи маски подсети.

Ответ: .

Для регистрации на сайте некоторой страны пользователю необходимо придумать пароль длиной ровно 11 символов. В пароле можно использовать десятичные цифры и 32 различных символа местного алфавита, причем все буквы используются в двух начертаниях – строчные и прописные. Каждый символ кодируется одинаковым и минимально возможным количеством бит, а каждый пароль – одинаковым и минимально возможным целым количеством байт. Определите объем памяти в байтах, необходимый для хранения 50 паролей.

**13**

Ответ: .

Исполнитель Чертёжник перемещается на координатной плоскости, оставляя след в виде линии. Чертёжник может выполнять команду Сместиться на (a, b) (где a, b – целые числа), перемещающую Чертёжника из точки с координатами (x, y) в точку с координатами (x + a, y + b).

Чертёжнику был дан для исполнения следующий алгоритм:

**14**

ТРЕНИРОВОЧНЫЙ КИМ № 170911

НАЧАЛО

Сместиться на (1, -3) ПОВТОРИ ... РАЗ

Сместиться на (..., …) Сместиться на (-1, -2) КОНЕЦ ПОВТОРИ

Сместиться на (-25, -33) КОНЕЦ

После выполнения этого алгоритма Чертёжник возвращается в исходную точку. Какое наибольшее число повторений могло быть указано в конструкции «ПОВТОРИ … РАЗ»?

Ответ: .

На рисунке изображена схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, З, И, К, Л, М. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город М, не проходящих через город Е?

**15**

# Б Д И Е

Г К

А М



З

В Ж Л

Ответ: .

Значение арифметического выражения 4911+733-7 записали в системе счисления с основанием 7. Сколько раз в этой записи встречается цифра 6?

**16**

Ответ: .

© 2018 Всероссийский проект «ЕГЭ 100 БАЛЛОВ» vk.com/ege100ballov Разбор всех заданий: vk.com/informatics\_100/2018kim01

 В таблице приведены запросы и количество страниц, которые нашел поисковый сервер по этим запросам в некотором сегменте Интернета:

 м c

ндексам т

9'3 а ни э ем н$$ авнь4 нн

і8м5с ив А c

9, 6

|  |  |
| --- | --- |
| Запрос | Количество страниц (тыс.) |
| Бабочка | 220 |
| Трактор | 400 |
| Fусеница | 360 |
| Трактор & Бабочка | 0 |
| Трактор & Fусеница | 160 |
| Трактор ЈГусеница ЈБабочка | 670 |

соответственно, т.е. A[0] = 4, A[1] = 7 и т.д. Определите значение переменной с после выполнения следующего фрагмента этой программы:

Сколько страниц (в тысячах) будет найдено по запросу Бабочка & Fусеница?

Ответ:

138 Укажите наибольшее целое положительное число Х, при котором истинно высказывание:

((Х - 1) < Х) ---г (40 > Х Х)

Ответ:

|  |  |
| --- | --- |
| Бейсик | Python |
| с = 0  FOR i = 1 ТО 9  IF А(i) A(0) THEN  с = с + 1 t = А(i)  А(i) = A(0) A(0) = t  END IF NEXT i | с = 0  for i in range(1,10): if А[i] А[0]:  с = с + 1  t = А[i] А[i] = A[0] A[0] = t |
| Алгоритмический язык | Паскаль |
| с := 0  нц для i от 1 до 9 еслн А[i] < A[0j то    t := А[i]  A[ij := А[0]  А[0] := t  все | с := 0;  for i := 1 to 9 do  if А[i] A[0j then begin с := с + 1;  t := А[i];  A[ij := А[0];  А[0] := t;  end; |
|  | |
| for (int i 1; i < 10; i++)  if (А[i] < А[0])( с++;  t = A[ij ; A[ij = A[0j; А[0] t; | |

Ответ:

Ы 2018 Всероссийский проект «ЕГЭ 100 БАЛЛОВ» vk.com/eqe100baIIov Разбор всех заданий: vk.com/informatics 100/2018kim01

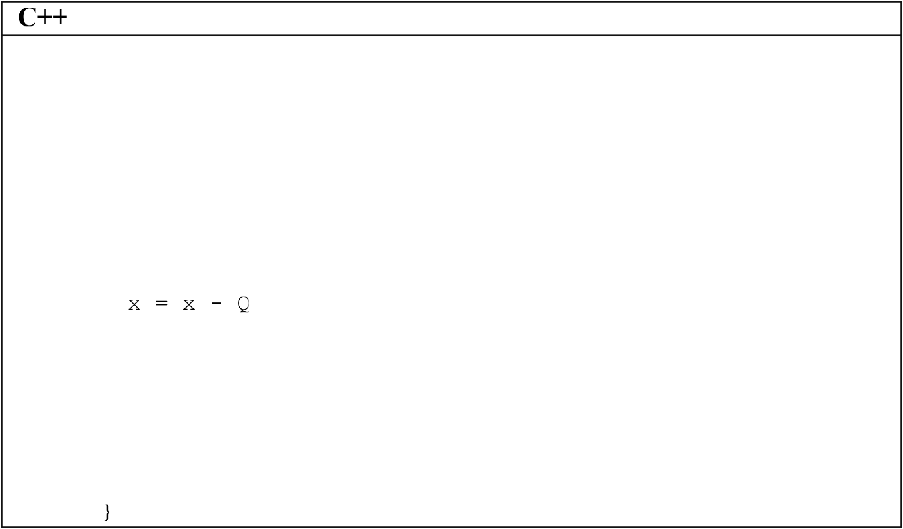
 HHme npHBepeH auropiiTM. VKam Te ari6onsiuee uHcno , npii BBope uoTOporo

iiiIFO]3HTM HaneuaTaeT criauana 3, riOTOM 5.





#include <iostream> using namespace std;

int main()(

int x, L, M, Q; cin >> x;

Q = 6;

L = 0;

while (x >= Q) { L = L 1;

if(Efl<L) (

M = L;

L = x



cout << L << endl << M << endl; return 0;

|  |  |
| --- | --- |
| **Seucxx** | Python |
| DIM X, L, M, Q AS INTEGER | x = int(input())  Q = 6   1. 0   while x >= Q:  L = L + 1   1. x   if M < L:    print(L) print(M) |
| INPUT X |
| Q 6 |
| L 0 |
| WHILE X >= Q |
| L L 1 |
| WEND |
| IF M L THEN |
| M L |
| L x |
| END IF |
| PRINT L |
| PRINT M |
| Anropu+uuuecx fi iiasix | Hacxans |
| apr  uau  uez x, L, M, Q  BBOQ X  Q := 6  L := 0  uu noma x >= Q L := L + 1      M x;  ecxu M < L  'I' O      ace  emrou L emrou M  KO H | var x, L, M, Q: integer;  begin  readln(x); Q := 6;  L := 0;  while x >= Q do begin L := L + 1;    end;    if M < L then begin M := L;  L := x;  end; writeIn(L); writeln(M);  end. |
|  | |

OTBeT:

21 Hanriiuii+e B o+Be+e uucuo, uOTopoe 6ype+ aneuaTa o B peayus+aTe Bbinon e z cueqyiotuero auropiiTMa.

|  |  |
| --- | --- |
| **Eeicxx** | Python |
| DIM A, B, T, M, R AS LONG | def F(x):  return 3\*(x\*x-16)\*(x\*x-16)+5 a = -10; b=20  M=a; R=F(a)  for t in range(a,b’1): if (F(t) < R):  M=t; R=F(t)  print (M\*R) |
| A = -10: B = 20 |
| M = A: R = F(A) |
| FOR T = A TO B |
| IF F(T) <R THEN |
| M = T |
| R = F(T) |
| END Ir |
| NEXT T |
| PRINT M’R |
| FUNCTION F (x) |



23 CKOnbKo pain **'IHbIX** peiue riii riMeeT cricTeMa nor uecKux ypaB eHHii

|  |  |
| --- | --- |
| Anropu+viuuecxuii uasix |  |
| apr  uau  uen a, b, t, M, R  a:=-10; b:=20  M:=a; R:=F(a)  ни для t от а до b если F(t) < R то  M:=t; R:=F(t)  Все  ameofl M’R  K O H  apr uex F(uen x)  nau  KON | var a,b,t,M,R :integer;  function r(x:integer):integer; begin  r : 3\*(x\*x-16)’(x\*x-16)’5; end;  begin  a := -10; b := 20;  M := a; R := F(a);  for t := a to b do begin if F(t) < R then begin  M : t;  R := F(t);  end; end;  write(M\*R); end. |
|  | |
| #include <iostream>  using namespace std;  long r(long x)(  return 3\*(x\*x-16)\*(x\*x-16)+5;  int main() (  long a -10, b = 20, M = a, R F(a); for (int t = a; t <= b; ”t) (  if (r(t)‹ R) (  M = t; R = F(t);  cout << M + R; return 0; | |

(xi x2 --+ xj) (x yi) = 1

(x xc X4) (x v y2) = 1

(X3 X4 x,) (x v y3) = 1

(x4 x, --+ X6) (x 4 y4) = 1

(x, X6 X7) (x, y,) = 1

roe xc, . . ., x6, y y—6

uorriuecxiie nepeMeHHnie? B xauecTBe oTBeTa HymHo





OTBeT:

yKazaTb KOu 'iecTBo TaKHx a6opOB.

OTBeT:

*He aadybome nepenecmu ace omaemui a 6nan« omaemoa X• 1 a coomaemcmauu c uncmpy«queii no a»inonnenum padomui.*

22 Hcnon riTeui• HIOHb 16 npeo6pa3yeT micro ma oxpa e. Y ucnon HTenz ecTb T]9Ii

**KoMaHpbl, KOTO]3I•IM** HpiicBoeribi rioMepa:

1. Hpii6a **BHTI>** I
2. **II)31f0aB44Ts** 2
3. YMHO **tI4T1• Hil** d

CKOnsKo cyuiecTByeT nporpaMM, ,I),iIII KOTOpsix npii IiGXopHOM 'iricne 2 pesynsTaToM sBnseTcs uricnO 16 **II H]3H 3TOM** TpaeKTopus Bbiuiicne riii we copepmiiT uiicnO 14?

OTBeT:



*9n» manner omaemoo ma aabanu» amen aacmu (24—27) venom Daytime E•HAHK OTBETOB X• 2. 3anumume ciiaaana iioMep sabanui (24, 25 u m. b.), a aameM neurone pemeiiue. Omaemci sanuc teatime aem«o u pazdopauao.*

24 Ha Bxop nporpaMMbi nocTynaeT aTypanb oe 'iiicno, we npeBbiiuaiotuee 10’.

Hydro HariHcaTb nporpaMMy, KoTopas BbIBO,£i,iiT ma oxpa MaKcuMansHyio

uuQpy uiicna, KpaTHyio 3. EGJIH B uucne eT uHQp, KpaT sIX 3, Tpe6yeTcz Ha oKpa BsiBecTii ‹ONO». B up BepeHHOii nporpaMMe ecTb OIIIII6KH.

|  |  |
| --- | --- |
| **Seficxx** | Pvthon |
| DIM N, D, M AS LONG  INPUT N M = 0  WHILE N > 0  D = N MOD 10  IF D MOD 3 = 0 THEN  Ir D > M THEN  M = D END IF  END IF  N = N 10 WEND  Ir M - O THEN PRINT "NO"  ELSE  PRINT M END IF | N = int(input())  m = 0  while N > 0: d = N % 10  if d % 3 == 0: if d > m:  m = d  N = N // 10  if m == 0: print("NO")  else:  print(m) |
| Anropu+uuuecxufi uai•ix | **Macxan&** |
| алг  Hdч  ueл N, d, m ввод N  m := 0  ни пока N > 0  d := mod(N,10)  если mod(d, 3) = 0 то если d > m то  m := d все  все  N := div(N,10)  КМ  ecex m = 0 io | var N, d, m: longint;  begin  readln(N); m := 0;  while N > 0 do begin d := N mod 10;  if d mod 3 = 0 then  if d > m then m := d; N := N div 10;  end;  if m = 0 then  writeln('NO') else writeln( m )  end. |

|  |  |
| --- | --- |
| swsou "NO" nuaue  ece  **KOH** |  |
|  | |
| #include <iostream>  using namespace std;  int main() {  long N, d, m; cin >> N;  m 0;  while (N > 0) ( d = N % 10;  if (d % 3 == 0) if (d > m)  m = d;  N = N / 10;  )  if (m == 0)  cout << "NO” << endl;  else  cout << m << endl; return 0; | |

ПоследоВаТельнО ВЬІНОЈІНиТе спедуюиіее.

1. Напиші4Те, чТО ВыВедеТ эТа програММа при ВВоде числа 170.
2. УКажі4Те наиМеньшее трёхзначное зНачеНие ВходНОй переМенной п, при ВВоде коТОрого програмМа ВыВедеТ праВильНый oTBeT. УКажиТе эТо oTBeT.
3. НайдиТе В програММе Все ошибКН (иХ МожеТ бы Tь одНа или нескольКо). Для Каждой ошибКll ВЬlпишиТе сТроку, В кОТОрОй она допущена, и приВедиТе эТу же cTpoKy В испраВпеііііоМ Виде.

 QaH іlелочисленный массиВ il3 30 элеменТоВ. ЭлемеНТЬі массиВа могут

приниМать иельІе ЗНачения ОТ 0 до 10 000 ВКЈlіОчиТельНО. Опишите на

**O@HOM ИЗ** i3hIKOB програММироВаНия алгориТМ, КОтО]ЗЬІй находиТ КОличестВО TpoeK подряд идущих элеМеНТоВ МассиВа, ТаКих, что средНее число paBHo произВедеНию КрайНиХ.

В КачесТВе резульТаТа необхОдиМО ВыВесТи количесТВо ТаКих TpoeK.

НаприМер, для МассиВа из шесТи эпеМенТоВ: 4 5 15 3 3 1 програММа должна

ВыВесТи 2 (подходяТ наборы 5 15 3 и 3 3 1).

 QBa i4гpoKa, Паша и Вапя играюТ В спедуюиіую игру. ЗадаН НекоТОрЬlй

|  |  |  |
| --- | --- | --- |
| **Beiicux** | Python | |
| CONST N AS INTEGER = 30 | # ponycKaeTcs Taxme |  |
| DIM A (1 TO N) AS LONG DIM I AS LONG,  J AS LONG, K AS LONG | # cnonn3oBaTn pBe  # iienou cneHHbie nepeueuHbie j  a = []  n = 30 | k |
| FOR I = 1 TO N  INPUT A(I) NEXT I | for i in range(0, n): a.append(int(input())) |  |
| END |  |  |
| Anropu+uuuecx fi ii3six | Hacxans | |
| apr  Has  uen N = 30 uenTa6 a[1:N] uen i, j, k  uu gos i OT 1 go N  BBop a[i]    **KOH** | const  N = 30;  var  a: array [1..N] of longint; i, j, k: longint;  begin  for i := 1 to N do read1n(a[i]);  end. | |
|  | | |
| #include <iostream>  using namespace std; const int N = 30;  int main() { long a[N]; long i, j, k;  for (i — 0; i<N; i++) cin >> a[i];  return 0; | | |

набор симВОЈlhнЬіх цепочек («слОВ»), В КОТОром ни O@HO GлOBo не яВляеТся

началоМ другогО. Игра НачинаеТся с пусТой GT]9OKII, В Конец коТОрой иг]ЗОКТІ по очереди дописі•іВаюТ букВі•і, по одной буКВе за ход Так, чТОбьІ полученная цепочка на КаждоМ шаге была НачалоМ ОдНОго из заданНі•ІХ слОВ. ПерВый ход делаеТ Паша. ВыигрыВаеТ ТоТ, КТо перВый сосТаВиТ **CJIOBO** из заданного набора.

Задание 1.

а) ОпределиТе, у кого из иFpOKoB есть Вhlигрьтшная страТегия для набора CJIOB (KATEP, КАЗАНТНН, КАЗАЧЕСТВО) .

6) Определите, у когО иЗ игрокоВ есТь ВЬіигрьтшНая сТратегия для Набора CJIOB (MACTEP. . . MACTEP, MACKA. . . MACKA }. В перВоМ СлоВе 377 раз НОВТоряеТся слОВО MACTEP, а Во ВТОЈЗОМ — 99 раз поВТоряеТся слОВО MACKA.

3apa e 2. B a6ope cnoB, npHBepeHHOM B Papa 1a, noMe siiTe MecTaMri pBe cocep rie 6yxBsl B JIIO6Ou cuoBe +aK, uTO6bI BbIur]3biui az cTpaTerris 6bina y ppyroro urpoKa.

3anauue 3. Qa a6op cnOB (FACTPHT, FACTPONH, FACHTENIa, HEPKA,

 HEPECT, HECVIIIKA, HECTOP}. V KOrO H3 HFpOKoB ecTb BbIliF]3biiu as

c+paTer s?

 На вход программы поступает последовательность из N целых положительных чисел, все числа в последовательности различны.

Рассматриваются все пары различных элементов последовательности (элементы пары не обязаны стоять в последовательности рядом, порядок элементов в паре не важен). Необходимо определить количество пар, для которых произведение элементов кратно 57.

**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 целое положительное число, не превышающее 10 000.

В качестве результата программа должна напечатать одно число: количество пар, в которых произведение элементов кратно 57.

*Пример входных данных.*

## 4

9

38

3

19

*Пример аыхоЬных Ьанных Ьля приаеЬённого аыше прммера входных Ьанных.*

4

*Пояснение.* Из четырёх заданных чисел можно составить 6 попарных произведений: 9 38, 9 3, 9 19, 38 3, 38 19, 3 19. Из них на 57 делятся 4

произведения (9 38; 9 19; 38 3; 3 19).

Требуется написать эффективную по времени и по памяти программу для решения описанной задачи.

Программа считается эффективной по времени, если при увеличении количества исходных чисел N в k раз время работы программы увеличивается не более чем в k раз.

Программа считается эффективной по памяти, если память, необходимая для хранения всех переменных программы, не превышает 1 Кбайт и не увеличивается с ростом N.

Максимальная оценка за правильную (не содержащую синтаксических ошибок и дающую правильный ответ при любых допустимых входных данных) программу, эффективную по времени и по памяти, — 4 балла.

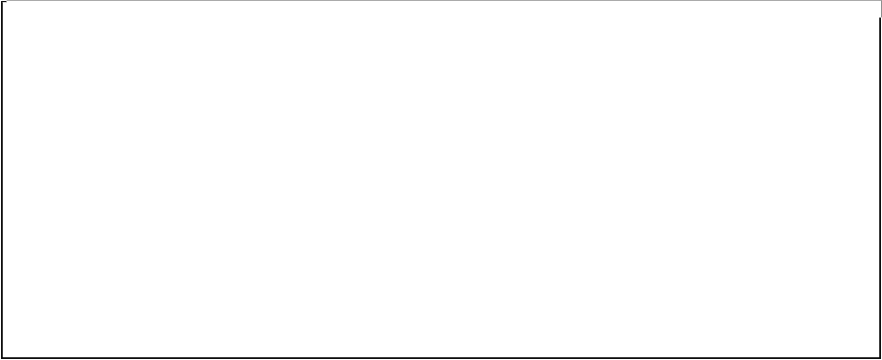
Максимальная оценка за правильную программу, эффективную только по времени— 3 балла.

Максимальная оценка за правильную программу, не удовлетворяющую требованиям эффективности, — 2 балла.

Вы можете сдать одну программу или две программы решения задачи (например, одна из программ может быть менее эффективна). Если Вы

сдадите две программы, то каждая из них будет оцениваться независимо от другой, итоговой станет большая из двух оценок.

Перед текстом программы обязательно кратко опишите алгоритм решения. Укажите используемый язык программирования и его версию.

О проекте «Пробный ЕГЭ каждую неделю»

Qанный ким составлен командой всероссийского волонтёрского проекта

«EFЭ 100 баллов» https://vk.com/eщe100ballov и безвозмездно распространяется для любых некоммерческих образовательных целей.

Нашли оши0ку в варианте?

Напишите нам, пожалуйста, и мы обязательно её исправим!

Для замечаний и **пожеланий:** https://vl‹.com/topic-10175642 35994595 (также **доступны** другие варианты для скачивания)

**Система оценивания экзаменационной** работы по информатике и ИКТ Часть 1

За правильный ответ на задания 1—23 ставится 1 балл; за неверный ответ или его отсутствие — 0 баллов.

|  |  |
| --- | --- |
| № задания | Ответ |
| 1 | 3 |
| 2 | ywzx |
| 3 | 13 |
| 4 | 4 |
| 5 | **11000** |
| 6 | 47 |
| 7 | 8 |
| 8 | 36 |
| 9 | 16 |
| 10 | 1876 |
| 11 | 7453 |
| 12 | 26 |
| 13 | **500** |
| 14 | 12 |
| 15 | 30 |
| 16 | 21 |
| 17 | **150** |
| 18 | 6 |
| 19 | 2 |
| 20 | 33 |
| 21 | -20 |
| 22 | 175 |
| 23 | **398** |

|  |  |
| --- | --- |
| **СОСТАВИТЕЛh ВАРИАНТА:** | |
| ФИО: | Войтюк Анна Владимировна |
| Предмет: | Информатика, математика |
| Стаж: | 10 лет |
| Регалии: | Организатор очных курсов подготовки к ЕГЭ по  информатике и математике |
| Аккаунт ВБ: | аппа voytyuk |
| Сайт и доп.  **информация:** | Занятия по подготовке к EFЭ по математике и  информатике в г.Краснодар  Запись на бесплатное тестирование по телефону  +79034549295 |

'tacos 2

24 Ha Bxop nporpaMMbi nocTynaeT aTypaub oe uiicno, se npeBbiiuaiotuee 10'.

Hyg o HariHcaTb nporpaMMy, KoTopas Bi•IBO,£i,iiT ca oxpa MaKcuMaub yio

uuQpy uiicna, upaTHyio 3. **EGJII4** B uricne **HeT** uiijip, xpaT **bIX** 3, Tpe6yeTcs ca

capa BsiBecTii ‹SNO». B npuBepeHHOii nporpaMue ecTb **OIIII46KH.**

|  |  |
| --- | --- |
| **Seucxx** | Python |
| DIM N, D, M AS LONG  INPUT N M = 0  WHILE N > 0  D = N MOD 10  IF D MOD 3 = 0 THEN IF D > M THEN  M D  END IF END IF  N = N 10 WEND  IF M = 0 THEN  PRINT "NO" ELSE  PRINT M END IF | N = int(input())  m = 0  while N > 0: d = N % 10  if d % 3 0:  if d > m: m = d  N = N // 10  if m == 0: print("NO")  else:  print(m) |
| Anropu+uuuecxuii uai•ix | **Macxan&** |
| алг  Hdч  ueл N, d, m ввод N  m := 0  ни пока N > 0  d : mod(N,10)  если mod(d, 3) 0 то если d > m то  m := d все  все  N := div(N,10)  КМ  eczw m = 0 so rmaor "NO"  uuaue  BbIB OQ ITU  ace  KOH | var N, d, m: longint;  begin  readln(N); m := 0;  while N > 0 do begin d := N mod 10;  if d mod 3 0 then  if d > m then m : d: N := N div 10;  end;  if m = 0 then  writeln('NO') else writeln( m )  end. |
| C++ | |
| #include <iostream>  using namespace std; | |

int main() (

long N, d, m; cin >> N;

while (N > 0) ( d = N % 10;

if (d % 3 == 0)

if (d > m)

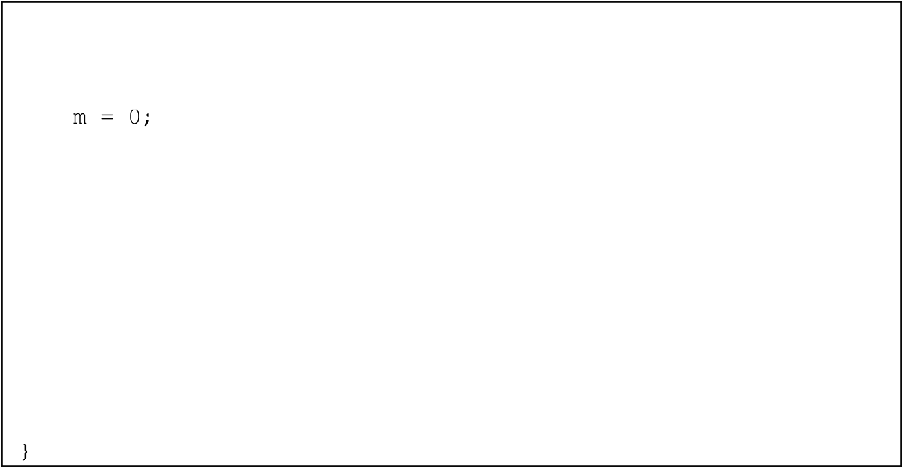
m = d;

N = N / 10;

)

else

cout << m << endl; return 0;

ПоследоВаТельнО ВыполниТе следующее.

if (m

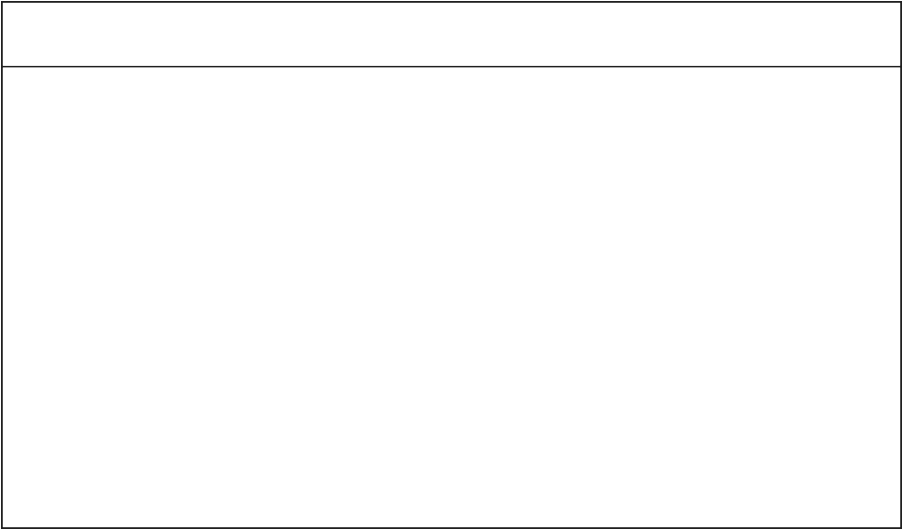
0)

cout << "NO" << endl;

1. НапишиТе, чТО ВыВедеТ эТа програМма при ВВоде числа 170.
2. УКажі4Те наиМеНьшее трёхзначНое значені4е Входной переМеННой n, при

ВВоде KoToporo програММа ВыВедеТ праВилЬНый oTBeT. УКажиТе эТо oTBeT.

1. Найдите В програМме Все ошибКlі (иХ МожеТ бы Tь одна ипи несКолькО). Дпя Каждой ошибКll ВЬlпишиТе строку, В кОТОрОй она допущена, и приВедиТе эТу же cTpoKy В испраВленноМ Виде.

Содержание верного отВета и указаНіія по оиениванию (допускаюТся иНые §зорМулирОВКі4 oTBeTa, не искажающие его сМыспа)

1. ПЈ9и ВВоде чгісла 170 програМма ВыВедеТ НеВерный oTBeT NO.
2. Наі4меНьшее ТрёхзначНое число, длІ КОТОрого пporpaмMa ВыдаёТ

праВипьНый oTBeT — 103.

1. ПрограММа содержиТ дВе оші4бКи:
2. НеВерное перВоначальное присВоеНие;
3. НеВерное услоВие при печаТи резульТаТа. Пример исправления для яЗыка Паскаль: ПерВая ошибКа:

m := 0;

ИспраВленная cTpoKa:

m := -1;

ВТорая ошибКа:

if m = 0 then ИспраВленНая сТрока: if m = -1 then

|  |  |
| --- | --- |
| *Пояснение Ьля эксперта*  При выводе m сравнивается с нулем, который должен сигнализировать о том, что нужных цифр (кратных 3) не нашли. Но 0 — это ведь тоже цифра, кратная 3, поэтому она может быть правильным ответом и использовать это значение для обнаружения отсутствия нужных цифр нельзя. Но если подходящая цифра — не 0, то вывод работает нормально. в качестве первоначального значения может быть использовано любое отрицательное число, например -1. | |
| Указания по оцениванию | Баллы |
| В задаче требуется выполнить три действия.   1. Указать результат программы при данном вводе.   Это действие считается выполненным, если указан верный результат работы программы при заданных входных данных. Экзаменуемый не обязан объяснять, как получен этот результат, достаточно указать верное число.   1. Указать минимальное трехзначное число, при котором программа выводит верный ответ.   Это действие считается выполненным, если указано правильное число. Ученик не обязан указывать, что будет выведено, и объяснять, как работает программа.   1. Найти и исправить ошибки в программе.   Это действие считается выполненным, если верно указаны обе ошибки и предложены верные варианты исправления, при этом никакие верные строки программы не указаны в качестве неверных. В исправленной строке допускаются незначительные синтаксические ошибки (лишние или пропущенные знаки препинания, неточные написания служебных слов языка). Ошибка считается исправленной, если выполнены оба следующих условия:  а) правильно указана строка с ошибкой;  6) указан такой новый вариант строки, что при исправлении другой ошибки получается правильная программа |  |
| Выполнены все три необходимых действия, и ни одна верная  строка не указана в качестве ошибочной |  |
| Не выполнены условия, позволяющие поставить 3 балла.  Имеет место одна из следующих ситуаций.   1. Выполнены два первых действия, найдена и исправлена одна ошибка в программе, ни одна верная строка не названа ошибочной. 2. Выполнены два первых действия, найдены и исправлены |  |

|  |  |
| --- | --- |
| две ошибки в программе, одна верная строка названа  ошибочной.  3. Выполнено одно из первых двух действий, найдены и исправлены две ошибки в программе, ни одна верная строка не названа ошибочной |  |
| Не выполнены условия, позволяющие поставить 2 или 3 балла.  При этом имеет место один из следующих случаев.   1. Выполнены два первых действия. При этом несущественно, насколько правильно выполнено третье действие. 2. Найдены и исправлены две ошибки в программе, не более чем одна верна» строка названа ошибочной. При этом несущественно, насколько правильно выполнены действия 1 и 2. 3. Выполнено одно из двух первых действий. Исправлена одна из двух ошибок. Не более чем одна верная строка названа ошибочной | 1 |
| Не выполнены условия, позволяющие поставить 1, 2 или 3  балла | 0 |
| *Максимальныи балл* | 3 |

25 Дан целочисленный массив из 30 элементов. Элементы массива могут принимать целые значения от 0 до 10 000 включительно. Опишите на

одном из языков программирования алгоритм, который находит количество троек подряд идущих элементов массива, таких, что среднее число равно произведению крайних.

В качестве результата необходимо вывести количество таких троек. Например, для массива из шести элементов: 4 5 15 3 3 1 программа должна

вывести 2 (подходят наборы 5 15 3 и 3 3 1).

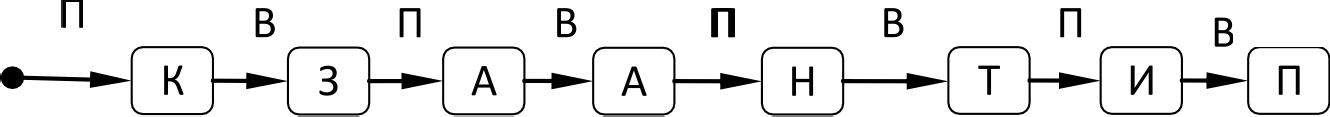
|  |  |
| --- | --- |
| **Бейсик** | Python |
| CONST N AS INTEGER = 30 | # допускается также |
| DIM А (1 ТО N) AS LONG | # использовать две |
| DIM I AS LONG, | # целочисленные переменные ј и k |
| Ј AS LONG, | а = [] |
| К AS LONG | п = 30 |
| FOR I = 1 ТО N | for i іп raпge(0, п): |
| INPUT А(I) NEXT I | a.append(int(input())) |
| END |  |
| Алгоритмический язык | Паскаль |
| алг | const |

|  |  |  |
| --- | --- | --- |
| нач  цел N = 30 целтаб a[1:N] цел i, ј, k  нц для i от 1 до N ввод а[i]  \*u | N = 30;  var  а: array [1..N] of longint; i, ј, k: longint;  begin  for i := 1 to N do readln(a[i]);  end. | |
|  | | |
| #include <iostream>  using namespace std; const int N = 30; int main() (  long a[Nj ; long i, ј, k;  for (i = 0; i<N; i++) cin >> a[ij ;  return 0; | | |
| Содержание верного ответа и указания по оцениванию  (допускаются иные формулировки ответа, не искажающие его смысла) | | |
| На языке Паскаль  k :— 0;  for i := 2 to N - 1 do  if(а[i] = a[i-1]\*a[i+1])) then iпc(k);  writeln(k);  На алгоритмическом языке  k := 0;  нц для i от 2 до N-1  если a[i]=a[i-1]\*a[i+1] то k := k+1  все кц  вывод k | | |
| **Указания по оцениванию** | |  |
| *Общие указания*   1. В алгоритме, записанном на языке программирования, допускается наличие отдельных синтаксических ошибок, не искажающих замысла автора программы. 2. Эффективность алгоритма не имеет значения и не | |  |

|  |  |
| --- | --- |
| оценивается.  3. Допускается запись алгоритма на языке программирования, отличном от языков, перечисленных в условии. В этом случае должны использоваться переменные, аналогичные описанным в условии. Если язык программирования использует типизированные переменные, описания переменных должны быть аналогичны описаниям переменных на естественном языке. Использование нетипизированных или необъявленных переменных возможно только в случае, если это допускается языком программирования; при этом количество переменных и их идентификаторы должны соответствовать условию задачи |  |
| Предложен правильный алгоритм, выдающий в качестве  результата верное значение | 2 |
| Не выполнены условия, позволяющие поставить 2 балла.  Предложено в целом верное решение, содержащее не более одной ошибки из числа следующих:   1. в цикле происходит выход за границу массива (например, при использовании цикла от 1 до N, от 2 go N или от 1 до N—1); 2. не инициализируется или неверно инициализируется счётчик количества найденных троек;   счётчик количества троек в цикле не изменяется или изменяется неверно;   1. при проверке выполнения условия набора элементов используются неверные индексы; 2. отсутствует вывод ответа; 3. используется переменная, не объявленная в разделе   описания переменных;   1. не указано или неверно указано условие завершения цикла; 2. индексная переменная в цикле не меняется (например, в цикле while) или меняется неверно; 3. неверно расставлены операторные скобки | 1 |
| Не выполнены условия, позволяющие поставить 1 или 2 балла | 0 |
| *Максимальныїі балл* | 2 |

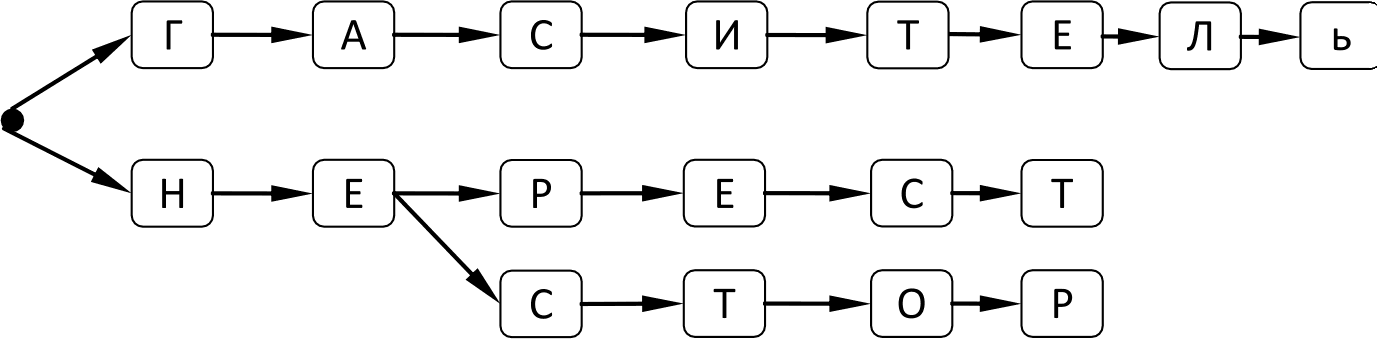
|  |  |
| --- | --- |
| 1. Чтобы выиграл Валя. нужно дать ему возможность выбрать слово на четном ходу. Например, если вместо КАЗАНТИП написать КЗААНТИП. то Валя может выбрать его на 2-ом ходу, и игра закончится его выигрышем за 8 шагов:      1. У Вали. Дерево всех возможных партий приводится на рисунке. Для Пети мы рассматриваем все возможные ходы, для Вали— только выигрышный вариант на каждом шаге. | |
| Указания по оцениванию |  |
| *Пре0варителсные замечания*  *В заЬаче от ученика требуется выполнить три задания. Их тру0ность аозрастает. Количество баллоа а целом соответствует количестау аыполненных заданий (подробнее см. ниже).*  Пункты la и 16 считаются выполненным, если (i) правильно указано, кто из игроков имеет выигрышную стратегию в на данном наборе, и (ii) описаны выигрышные стратегии — так, как это сделано в образце решения, или любым другим способом (таблица, словесно и т.д.). Первое задание считается выполненным полностью, если выполнены полностью оба пункта: la и 16.  Замечание для проверяющего. Описать стратегию игрока — значит описать, какой ход он должен сделать в любой ситуации, которая ему может встретиться при различной игре противника (см. условие задачи). Экзаменуемый может описывать стратегию любым удобным ему способом.  Существенно, чтобы было понятно, какой ход должен сделать игрок, реализующий стратегию, и было показано, что все |  |

1. Два игрока, Паша и Валя играют в следующую игру. Задан некоторый набор символьных цепочек («слов»), в котором ни одно слово не является

началом другого. Игра начинается с пустой строки, в конец которой игроки по очереди дописывают буквы, по одной букве за ход так, чтобы полученная цепочка на каждом шаге была началом одного из заданных слов. Первый ход делает Паша. Выигрывает тот, кто первый составит слово из заданного набора.

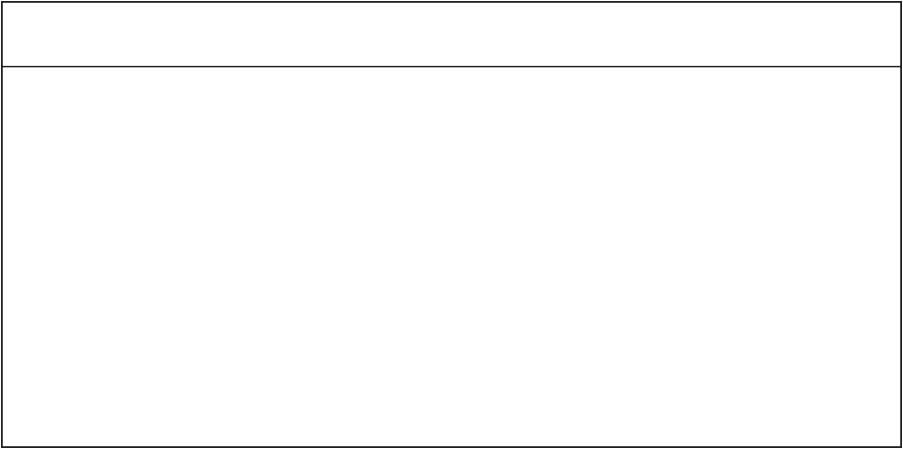
Задание 1.

а) Определите, у кого из игроков есть выигрышная стратегия для набора слов (KATEP, КАЗАНТИП, КАЗАЧЕСТВО}.

* 1. Определите, у кого из игроков есть выигрышная стратегия для набора слов (MACTEP... MACTEP, MACKA... MACKA }. В первом слове 377 раз повторяется слово MACTEP, а во втором — 99 раз повторяется слово MACKA.

Задание 2. В наборе слов, приведённом в задании la, поменяйте местами две соседние буквы в любом слове так, чтобы выигрышная стратегия была у другого игрока.

Задание 3. Дан набор слов (ГАСТРИТ, ГАСТРОЛИ, ГАСИТЕЛЬ, HEPKA, HEPECT, НЕСУШКА, HECTOP} . V кого из игроков есть выигрышная стратегия?

Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)

* + 1. а) У Паши. Выбор слова возможен на 3-ем ходу игры. 3-ий ход будет делать Паша, у него есть возможность выбрать слово KATEP. Тогда игра закончится на 5-м ходу, и Паша выиграет. Выигрышная стратегия:

в п в п

## К А Т Е Р

6) У Вали. Ходы с нечетными номерами делает Петя, с четными — Валя. Выбор слова происходит на 4-ом ходу, таким образом выбор делает Валя. Первое слово содержит 6\*377 букв, это четное количество. Выбрав первое слово, Валя сделает последний ход и выиграет.

В первой строке входных данных задаётся количество чисел N (1 < N < 1000). В каждой из последующих N строк записано одно целое положительное число, не превышающее 10 000.

|  |  |
| --- | --- |
| возможные заключительные позиции выигрышные для этого  игрока.  Задание 2 считается выполненным, если правильно указано изменённое слово, и указано, кто из игроков имеет выигрышную стратегию и как именно он может ее реализовать.  Задание 3 считается выполненным, если (i) правильно указано, что выигрышную стратегию имеет Валя; (ii) правильно описано дерево всех партий, возможных при этой выигрышной стратегии (в виде рисунка или таблицы). Во всех случаях стратегии могут быть описаны так, как это сделано в примере решения, или другим способом. |  |
| Выполнены все три задания.  Здесь и далее в решениях допускаются описки, которые не искажают сути решения и не приводят к неправильному ответу |  |
| Не выполнены условия, позволяющие поставить 3 балла, и  выполнено одно из следующих условий.   * Выполнено третье задание. * Выполнены первое и второе задания. | 2 |
| Не выполнены условия, позволяющие поставить 3 или 2 балла,  и выполнено одно из следующих условий.   * Первое задание выполнено, * Второе задание выполнено | 1 |
| Не выполнено ни одно из условий, позволяющих поставить 1,  2 или 3 балла | 0 |
| *Максимальный балл* | 3 |

В качестве результата программа должна напечатать одно число: количество пар, в которых произведение элементов кратно 57.

*Пример входных данных.*

4

9

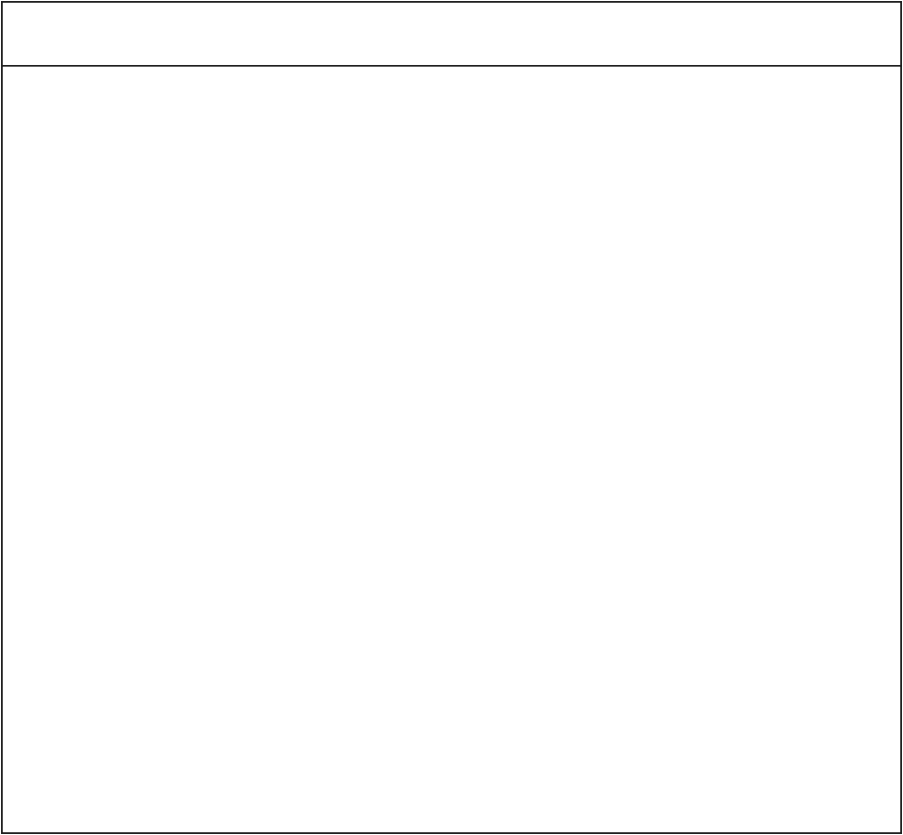
38

3

19

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

4

Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)

Произведение двух чисел делится на 57, если выполнено одно из следующих условий (условия не могут выполняться одновременно). А. Оба сомножителя делятся на 57.

fi. Один из сомножителей делится на 57, а другой не делится.

В. Ни один из сомножителей не делится на 57, но один сомножитель делится на 3, а другой— на 19.

1. На вход программы поступает последовательность из N ііелых положительных чисел, все числа в последовательности различны.

Рассматриваются все пары различных элементов последовательности (элементы пары не обязаны стоять в последовательности рядом, порядок элементов в паре не важен). Необходимо определить количество пар, для которых произведение элементов кратно 57.

**Описание** входных и выходных **данных**

*Примечание Ьля проаеряюіцего.*

Условие делимости произведения на 57 можно сформулировать проще, например, так: (один из сомножителей делится на 57) ИЛИ (один сомножитель делится на 3, а другой — на 19). Но в этом случае пapa сомножителей может удовлетворять обоим условиям, что затруднит подсчёт количества пар. При вводе чисел можно определять, делится ли каждое из них на 57, 3 и 19, и подсчитывать следующие значения:

1. п57 — количество чисел, кратных 57;
2. пЗ — количество чисел, кратных 3, но не кратных 57;
3. п19 — количество чисел, кратных 19, но не кратных 57.

*Примечание Оля проверяющего.*

Сами числа при этом можно не хранить. Каждое число учитывается не более чем в одном из счётчиков. Количество пар, удовлетворяющих условию А, можно вычислить по формуле п57 (п5—7 1)/2. Количество пар, удовлетворяющих условию Б, можно вычислить по формуле п57 (N— п57). Количество пар, удовлетворяющих условию В, можно вычислить по формуле пЗ п19. Поэтому искомое количество пар вычисляется по формуле

п57 (п57 1)/2 + п57’(N — п57) + пЗ пl9.

*Ниже npиaebeнa реализующая описанный алгоритм прогртяма на языке Паскалс (использована аерсия PascalABC)*

var N: integer; (количество чисел} а: integer; (очередное число} п57, п19, пЗ: integer;

k57: integer; (количество требуемых пар} i: integer;

begin readln(N);

п57:=0; п19:=0; п3.=0;

for i:=1 to N do begin readln(a);

if а mod 57 = 0 then п57 := n57+1

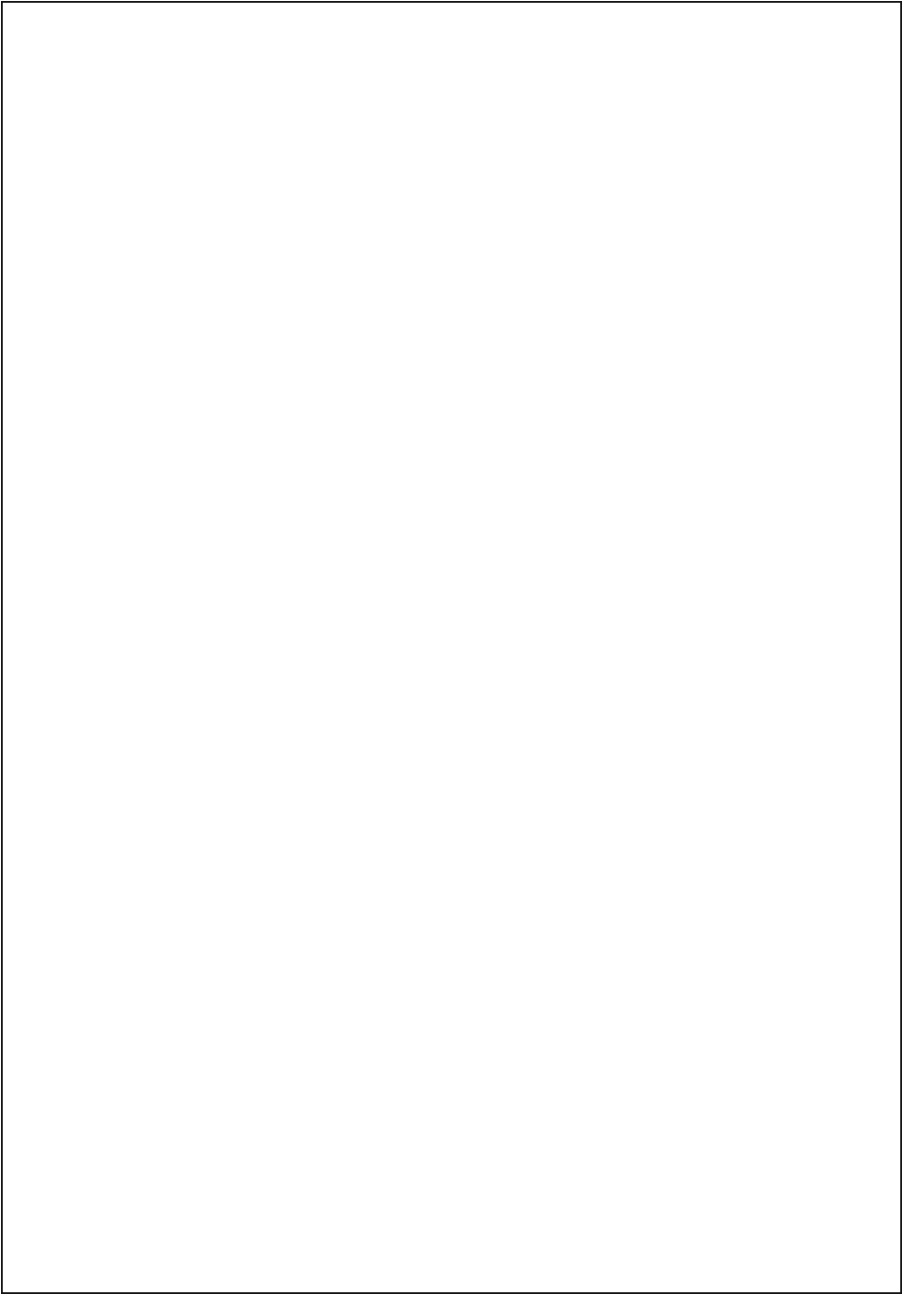
else if а mod 19 = 0 then п19 := п19 + 1

else if а mod 3 = 0 then пЗ := пЗ + 1;

end.

end;

k57 := п57\*(п57-1) div 2 + n57\*(N-n57) + пЗ\*п19; writeln(k57)



|  |  |
| --- | --- |
| Решение, не эффективное ни по времени, ни по памяти, запоминает  входную последовательность в массиве, после чего явно перебирает все возможные пары. Такое решение оценивается в 2 балла | |
| Указания по оцениванию | Баллы |
| Если в работе представлены две программы решения задачи,  то каждая из них независимо оценивается по указанным ниже критериям, итоговой считается большая из двух оценок.  Описание алгоритма решения без программы не оценивается.  Общий принцип оценивания можно неформально описать так. Эффективная правильная программа (возможно, с небольшим количеством синтаксических ошибок, подробнее см. ниже в критериях) оценивается 4 баллами.  1 балл снимается за наличие одной содержательной ошибки (примерный список ошибок см. ниже в критериях).  1 балл снимается за хранение исходных данных в массиве или другой аналогичной структуре, размер которой растёт с ростом количества элементов N. | 4 |
| Программа правильно работает для любых входных данных  произвольного размера. Используемая память не зависит от количества прочитанных чисел, а время работы пропорционально этому количеству.  Допускается наличие в тексте программы до трёх синтаксических ошибок одного из следующих видов:   1. пропущен или неверно указан знак пунктуации; 2. неверно написано или пропущено зарезервированное слово языка программирования; 3. не описана или неверно описана переменная; 4. применяется операция, не допустимая для соответствующего типа данных. | 4 |

Ещё один возможный вариант (есть и другие) — подсчёт количества чисел, которые не делятся на 57, — можно вести по формуле пЗ+п19+пх, где nx количество чисел, которые не делятся ни на 3, ни на 19. Значение nx можно вычислить с помощью отдельного счётчика.

Все подобные программы оцениваются в 4 балла.

При любом наборе вспомогательных величин возможны различные способы записи итоговой формулы. Можно, например, раскрывать скобки и приводить подобные члены или, наоборот, выносить за скобки общие множители; можно вводить дополнительные переменные для отдельных слагаемых, а затем вычислять их сумму.

Допустим любой способ записи вычислений, эквивалентный правильной формуле, выбранный способ записи не влияет на оценку.

Возможно решение, основанное на описанных идеях, однако предварительно сохраняющее элементы последовательности в массив. Такое решение (если в нём нет ошибок) эффективно по времени, но неэффективно по памяти. Оно оценивается в 3 балла.

|  |  |
| --- | --- |
| Если одна и та же ошибка встречается несколько раз, это  считается за одну ошибку. |  |
| Не выполнены условия, позволяющие поставить 4 балла.  Программа в целом работает правильно для любых входных данных произвольного размера.  Время работы пропорционально количеству введённых чисел; правильно указано, какие величины должны вычисляться по ходу чтения элементов последовательности чисел.  Количество синтаксических ошибок («описок») указанных в критериях на 4 балла, не более пяти.  Допускается наличие не более одной ошибки следующих видов:   1. допущена ошибка при вводе данных, например, не считывается значение N или числа могут быть считаны, только если будут записаны в одной строке через пробел; 2. ошибка при инициализации или отсутствие инициализации счётчиков; 3. в программе перепутаны знак целочисленного деления и взятия остатка или знаки операций «равно» и «не равно», «or» вместо «and» и т.п.; 4. использована неверная структура проверок, в результате которой некоторые счётчики могут получить неверное значение; 5. получены правильные значения счётчиков (вспомогательных величин), которые в принципе позволяют получить требуемое количество пар, но формула для вычисления записана неверно (комбинаторная ошибка); 6. отсутствует вывод ответа, или выводится значение не той переменной; 7. в описании алгоритма правильно описан смысл используемых вспомогательных величин, и в программе правильно записан алгоритм вычисления искомого количества пар, исходя из этих величин, однако при вычислении одной из вспомогательных величин допущена ошибка.   3 балла также ставится за программу, в которой нет содержательных ошибок, но используемая память зависит от количества прочитанных чисел (например, входные данные запоминаются в массиве, контейнере STL в С++ или другой |  |

|  |  |
| --- | --- |
| аналогичной структуре данных). |  |
| Не выполнены условия, позволяющие поставить 3 или 4 балла.  Программа работает в целом верно, эффективно или нет.  В реализации алгоритма допускается до трёх содержательных ошибок, описанных в критериях на 3 балла.  Количество синтаксических ошибок, указанных в критериях на 4 балла, не должно быть более девяти.  2 балла также ставится за корректное переборное решение, в котором все числа сохраняются в массиве (или другой аналогичной структуре), рассматриваются все возможные пары и подсчитывается количество подходящих произведений. Пример фрагмента соответствующей программы на языке Паскаль:  k := 0;  for i := 1 to п - 1 do for ј := i + 1 to п do  if a[i]\*a[j] mod 57 = 0 then k := k + 1; writeln(k);  В реализации переборного алгоритма не допускаются логические ошибки, например, когда учитываются произведения вида а[i]\*а[i] или пары считаются дважды. | 2 |
| Не выполнены условия, позволяющие поставить 2, 3 или 4  балла.  Из описания алгоритма и общей структуры программы видно, что экзаменуемый в целом правильно представляет путь решения задачи.  Допускается любое количество «описок».  1 балл ставится также за решения, верные лишь в частных случаях | 1 |
| Не выполнены критерии, позволяющие поставить 1, 2, 3 или 4  балла | 0 |
| *Максимальныїі балл* | 4 |