

ВАРИАНТ 2

Задание 2.1

Даны натуральные числа A и B . Требуется найти такое минимально возможное натуральное число $K \geq A$, что сумма всех чисел от A до K больше B . 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 |
|---|---|
| <pre>DIM A,B,S,K AS INTEGER INPUT A,B S = 0 K = A WHILE S < B K = K + 1 S = S + K WEND PRINT K END</pre> | <pre>a = int(input()) b = int(input()) s = 0 k = a while s < b: k = k + 1 s = s + k print(k)</pre> |
| Алгоритмический язык | Паскаль |
| <pre>алг нач цел a, b, s, k ввод a, b s := 0 k := a нц пока s < b k := k+1 s := s+k кц вывод k кон</pre> | <pre>var a, b, s, k: integer; begin read(a,b); s := 0; k := a; while s < b do begin k := k+1; s := s+k; end; writeln(k) end.</pre> |
| Си | |
| <pre>#include <stdio.h> int main(){ int a, b, s, k; scanf("%d %d", &a, &b); s = 0; k = a; while (s < b) { k = k+1; s = s+k; } printf("%d", k); return 0; }</pre> | |

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе чисел 15 и 26.
2. Приведите пример значений A и B , при вводе которых программа выведет верный ответ. Укажите этот ответ.
3. Найдите в программе все ошибки (их может быть одна или несколько). 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.

Содержание верного ответа

(допускаются иные формулировки ответа, не искажающие его смысла)

1. При вводе чисел 15 и 26 программа выведет число 17.
2. Примеры чисел, при вводе которых программа выводит верный ответ: 3 и 8 (ответ 5), 10 и 22 (ответ 12), 4 и 16 (ответ 7).
3. Программа содержит две ошибки:
 - 1) неверная инициализация;
 - 2) неверное условие цикла.

Пример исправления для языка Паскаль:

Первая ошибка:

`s := 0;`

Исправленная строка:

`s := a;`

Возможен и другой вариант исправления, приводящий к верному результату.

Ошибочная строка:

`k := a;`

Исправленная строка:

`k := a-1;`

В данном случае начальные значения переменных *s* и *k* не согласованы между собой, исправить можно любое из них.

Вторая ошибка:

`while s< b do begin`

Исправленная строка:

`while s<= b do begin`

В программах на других языках ошибочные строки и их исправления аналогичны.

Незначительной опечаткой, не влияющей на оценку, следует считать отсутствие знаков и служебных слов после содержательной части исправления.

Задание 2.2

Дан массив, содержащий 2016 положительных целых чисел, не превышающих 1000. Необходимо найти и вывести максимальный из тех элементов этого массива, шестнадцатеричная запись которых содержит ровно 2 цифры. Если таких чисел в массиве нет, ответ считается равным нулю. Исходные данные объявлены так, как показано ниже. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать часть из описанных.

| Бейсик | Python |
|--|--|
| CONST N=2016 DIM A(N) AS INTEGER DIM I, M, K AS INTEGER FOR I = 1 TO N INPUT A(I) NEXT I ... END | # допускается также использование # целочисленных переменных m, k a = [] N = 2016 for i in range(0, N): a.append(int(input())) ... |
| Алгоритмический язык | Паскаль |
| <u>алг</u> <u>нач</u> <u>цел</u> N=2016 <u>целтаб</u> a[1:N] <u>цел</u> i, m, k <u>нц для</u> i <u>от</u> 1 <u>до</u> N <u>ввод</u> a[i] <u>кц</u> ... <u>кон</u> | const N=2016; var a: array [1..N] of integer; i, m, k: integer; begin for i:=1 to N do readln(a[i]); ... end. |
| Си | |
| #include <stdio.h> #define N 2016 int main(){ int a[N]; int i, m, k; for (i=0; i<N; i++) scanf("%d", &a[i]); ... return 0; } | |

Содержание верного ответа

(допускаются иные формулировки ответа, не искажающие его смысла)

В системе счисления с основанием p две цифры требуется для записи чисел в диапазоне $[p; p^2)$. При $p = 16$ получаем диапазон $[16; 256)$. Для решения задачи необходимо просмотреть все числа в массиве и выбрать максимальное среди тех, которые попадают в этот диапазон

Пример правильной программы на языке Паскаль

```
m:=0;  
for i:=1 to N do begin  
    if (16<=a[i]) and (a[i]<256) and (a[i]>m)  
        then m := a[i];  
end;  
writeln(m)
```

Задание 2.3.

Два игрока, Паша и Валя, 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 **два** камня или увеличить количество камней в куче в **три** раза. Например, имея кучу из 15 камней, за один ход можно получить кучу из 17 или 45 камней. У каждого игрока, чтобы делать ходы, есть неограниченное количество камней. Игра завершается в тот момент, когда количество камней в куче становится не менее 36. Если при этом в куче оказалось не более 85 камней, то победителем считается игрок, сделавший последний ход. В противном случае победителем становится его противник. Например, если в куче было 30 камней и Паша утроит количество камней в куче, то игра закончится и победителем будет Валя. В начальный момент в куче было S камней, $1 \leq S \leq 35$.

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. Выполните следующие задания.

1. а) При каких значениях числа S Паша может выиграть в один ход?
Укажите все такие значения и соответствующие ходы Паши.
б) У кого из игроков есть выигрышная стратегия при $S = 28, 30, 32$?
Опишите выигрышные стратегии для этих случаев.
2. У кого из игроков есть выигрышная стратегия при $S = 10, 8$? Опишите соответствующие выигрышные стратегии.
3. У кого из игроков есть выигрышная стратегия при $S = 6$? Постройте дерево всех партий, возможных при этой выигрышной стратегии (в виде рисунка или таблицы). На ребрах дерева указывайте, кто делает ход, в узлах – количество камней в позиции.

Содержание верного ответа

(допускаются иные формулировки ответа, не искажающие его смысла)

1. а) Петя может выиграть, если $S = 12, 13, \dots, 28, 34, 35$. При $S = 34$ и $S = 35$ первым ходом нужно добавить в кучу 2 камня, при остальных указанных значениях S нужно утроить количество камней.

б) При $S = 28$ Паша выигрывает в один ход, утраивая количество камней (см. п. а). При $S = 30$ или 32 утраивать количество камней не имеет смысла, т. к. после такого хода выигрывает противник. Поэтому можно считать, что единственный возможный ход – это добавление в кучу двух камней. При $S = 32$ после такого хода Паши в куче станет 34 камня. В этой позиции ходящий (то есть, Валя) выигрывает (см. п. а)). То есть, при $S = 32$ Паша (игрок, который должен ходить первым) проигрывает. Выигрышная стратегия есть у Вали. При $S = 30$, после того как Паша своим первым ходом добавит два камня, в куче станет 32 камня. В этой позиции ходящий (то есть Валя) проигрывает (см. выше). То есть при $S = 30$ Паша (игрок, который должен ходить первым) выигрывает. Выигрышная стратегия есть у Паши.

Замечание для проверяющего. Скорее всего, решение экзаменуемого будет не столь подробным. Это не является ошибкой. Ученик может, например, нарисовать деревья всех возможных партий для указанных значений S . Другая возможность – (1) указать на то, что при $S = 32$ и 30 утраивать кучу смысла не имеет, и (2) последовательно сводить случай $S = 32$ к случаю $S = 34$, а случай $S = 30$ – к случаю $S = 32$.

2. При $S = 10$ после первого хода Паши в куче будет либо 12, либо 30 камней. В обоих случаях выигрышная стратегия есть у игрока, который должен ходить, теперь это Валя. Случай $S = 12$ рассмотрен в задании 1а, а случай $S = 30$ – в задании 1б. Поэтому выигрышная стратегия есть у Вали. При $S = 8$ выигрышная стратегия есть у Паши. Ему нужно первым ходом добавить 2 камня и получить кучу из 10 камней. Как показано выше, в этой ситуации выигрышная стратегия есть у игрока, который НЕ должен ходить, то есть у Паши.

3. При $S = 6$ выигрышная стратегия есть у Вали. После первого хода Паши в куче может

стать либо 8, либо 18 камней. В обеих этих позициях выигрывает игрок, который будет делать ход (теперь это Валя). Случай $S = 8$ рассмотрен в п. 2, случай $S = 18$ рассмотрен в п. 1а.

В таблице изображено дерево возможных партий при описанной стратегии Вали. Заключительные позиции (в них выигрывает Валя) подчёркнуты. На рисунке это же дерево изображено в графическом виде (оба способа изображения дерева допустимы).

| Положения после очередных ходов | | | | | | |
|---------------------------------|-------------------------|--|-------------------------|--|-------------------------|--|
| Исх. положение | 1-й ход Паши (все ходы) | 1-й ход Вали (только ход по стратегии) | 2-й ход Паши (все ходы) | 2-й ход Вали (только ход по стратегии) | 3-й ход Паши (все ходы) | 3-й ход Вали (только ход по стратегии) |
| 6 | $6+2=8$ | $8+2=10$ | $10*3=30$ | $30+2=32$ | $32+2=34$ | $34+2=36$ |
| | | | | | $32*3=96$ | |
| | | | $10+2=12$ | $12*3=36$ | | |
| | $6*3=18$ | $18*3=54$ | | | | |

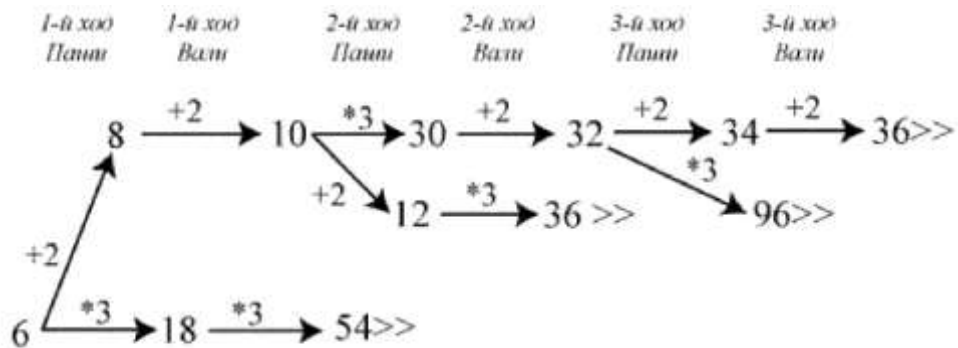


Рис.1. Дерево всех партий, возможных при Валиной стратегии.
 Знаком обозначены позиции, в которых партия заканчивается

Задание 2.4

Дан набор из N целых положительных чисел. Необходимо определить, какая цифра чаще всего встречается в десятичной записи чисел этого набора. Если таких цифр несколько, необходимо вывести наибольшую из них. Напишите эффективную по времени и по памяти программу для решения этой 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 одного килобайта 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 или только по памяти, – 3 балла.

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

Вы можете сдать **одну** или **две** программы решения задачи. Если Вы сдадите две

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

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

Описание входных и выходных данных

В первой строке входных данных задаётся количество чисел N ($1 \leq N \leq 1000$).

В каждой из последующих N строк записано одно натуральное число, не превышающее 10000.

Пример входных данных:

3
15
25
32

Пример выходных данных для приведённого выше примера входных данных:

5

В десятичной записи чисел заданного набора чаще всего – по 2 раза – встречаются цифры 2 и 5, большая из них – 5.

Содержание верного ответа

(допускаются иные формулировки ответа, не искажающие его смысла)

Необходимо создать массив из 10 элементов с индексами от 0 до 9 и использовать его для подсчёта количества всех цифр. Использование массива не делает программу неэффективной по памяти, так как размер массива не зависит от N . Затем нужно найти значение максимального элемента этого массива и вывести индекс этого элемента. Вместо массива можно использовать 10 отдельных переменных. В этом случае программа остаётся правильной и эффективной, но становится очень громоздкой, повышается вероятность ошибки. Если в проверяемой работе использован такой способ, эту работу следует проверить особенно тщательно.

Ниже приведена реализующая описанный выше алгоритм программа на языке Паскаль (использована версия Pascal ABC)

Пример правильной и эффективной программы на языке Паскаль

```
var
N: integer; {количество чисел}
a: integer; {очередное число}
digit: integer; {цифра числа}
d: array [0..9] of integer; {подсчёт цифр}
dmx: integer; {самая частая цифра}
i: integer;
begin
  for i:=0 to 9 do d[i]:=0;
  readln(N);
  for i:=1 to N do begin
    readln(a);
    while a>0 do begin
      digit := a mod 10;
      d[digit] := d[digit] + 1;
      a := div 10
    end;
  end;
  dmx := 0;
  for i:=1 to 9 do begin
    if d[i] >= d[dmx] then dmx := i;
  end;
  writeln(dmx);
end.
```