****

**ВСЕРОССИЙСКАЯ ОЛИМПИАДА ШКОЛЬНИКОВ ПО ИНФОРМАТИКЕ И ИКТ**

ШКОЛЬНЫЙ ЭТАП 2020–2021 УЧЕБНЫЙ ГОД

**РЕШЕНИЯ ШКОЛЬНОГО ЭТАПА**

**Задания, решения и критерии оценивания для школьного тура олимпиады по информатике и ИКТ, 9-11 класс**

**2020 – 2021 год**

**Максимальный балл за шесть задач: 600 баллов**

Входные и выходные данные должны соответствовать требованию (учитывая ввод(вывод) в одну строку или несколько). Неправильный формат ввода оценивается в 0 баллов, неправильный формат вывода оценивается в 0 баллов.

**Задача 1. Рекет на дороге**

**Максимальный балл: 100**

**Данные вводятся с клавиатуры или из файла input.txt, выводятся на экран или в файл output.txt.**

**Условие**

Карабас Барабас выдал Буратино несколько монет, чтобы он отнес их папе Карло. На дороге к дому Буратино подстерегают кот Базилио и лиса Алиса, которые согласны пропустить Буратино за 10 монет. Дуремар же согласен провести Буратино к дому по другой дороге за половину монет, которые Карабас Барабас выдал Буратино. Требуется определить сколько монет сможет донести Буратино до папы Карло.

**Формат входных данных**

Вводится одно четное число, не превосходящее100 — количество монет, которые выдал Карабас Барабас.

**Формат выходных данных**

Программа должна вывестиодно число — количество монет, которые Буратино сможет донести до папы Карло.

**Пример входных и выходных данных**

|  |  |  |
| --- | --- | --- |
| **Входные данные** | **Выходные данные** | **Комментарий** |
| 12 | 6 | Если Карабас Барабас выдал 12 монет, то выгоднее отдать половину монет (6 штук) Дуремару, чем 10 монет коту Базилио и лисе Алисе. В этом случае Буратино сможет донести до папы Карло 12-6=6 пирожков. |
| 100 | 90 | Если Карабас Барабас выдал 100 монет, то выгоднее отдать 10 монет коту Базилио и лисе Алисе, чем половину (50 монет) Дуремару. До папы Карло в этом случае Буратино донесет 100-10=90 монет. |
| 20 | 10 | Если выдано 20 монет, то в любом случае (и если отдать половину монет Дуремару, и если отдать 10 монет коту Базилио и лисе Алисе) папе Карло останется 10 монет. |

**Оценка задачи**

**Жюри 10 раз тестирует программу, один тест оценивается 10 баллами.**

Решение и критерии оценивания

**Пример входных и выходных данных для жюри**

|  |  |  |  |
| --- | --- | --- | --- |
| **№ теста** | **Входные данные** | **Выходные данные** | **Баллы** |
| 1 | 12 | 6 | 10 |
| 2 | 100 | 90 | 10 |
| 3 | 90 | 10 | 10 |
| 4 | 80 | 70 | 10 |
| 5 | 70 | 60 | 10 |
| 6 | 60 | 50 | 10 |
| 7 | 50 | 40 | 10 |
| 8 | 40 | 30 | 10 |
| 9 | 30 | 20 | 10 |
| 10 | 20 | 10 | 10 |
| **Итого** | | | **100** |

**Возможный вариант решения задачи на языке Pascal:**

program task\_2m;

var k,p: integer;

begin

assign(input,'input.txt'); reset(input);

assign(output,'output.txt'); rewrite(output);

readln(k);

p:= k div 2;

if p > 9 then writeln(k-10)

elsewriteln(p)

end.

**Возможный вариант решения задачи на языке С++:**

#include<iostream>

using namespace std;

int main()

{

int n;

cin>>n;

cout<<max(n/2,n-10)<<endl;

system ("\npause");

}

**Задача 2. Повышение уровня**

**Максимальный балл: 100**

**Данные вводятся с клавиатуры или из файла input.txt, выводятся на экран или в файл output.txt.**

**Условие**

Повышения уровня h в компьютерной игре зависит от количества набранного опыта d при выполнении заданий. Количество опыта необходимого для перехода на следующий уровень равно произведению положительных множителей вида (h – k\*d), где k=0, 1, 2, ...

Помогите игрокам вычислить сколько понадобится опыта для перехода на новый уровень в зависимости от номера текущего уровня и имеющегося опыта игрока.

**Формат входных данных**

В одной строке сначала вводится целое числоh и через один пробел целое число d.

**Формат выходных данных**

Выведите одно натуральное число – количество опыта для перехода на новый уровень.

**Ограничения**

1 ≤ h ≤ 10

1 ≤ d ≤ 100

**Пример входных и выходных данных**

|  |  |
| --- | --- |
| **Входные данные** | **Выходные данные** |
| 9 2 | 945 |
| 8 2 | 384 |

**Оценка задачи**

**Жюри 10 раз тестирует программу, один тест оценивается 10 баллами.**

Решение и критерии оценивания

**Пример входных и выходных данных для жюри**

|  |  |  |  |
| --- | --- | --- | --- |
| **№ теста** | **Входные данные** | **Выходные данные** | **Баллы** |
| 1 | 9 2 | 945 | 10 |
| 2 | 8 2 | 384 | 10 |
| 3 | 8 3 | 80 | 10 |
| 4 | 5 15 | 5 | 10 |
| 5 | 9 4 | 45 | 10 |
| 6 | 9 1 | 362880 | 10 |
| 7 | 8 1 | 40320 | 10 |
| 8 | 7 2 | 105 | 10 |
| 9 | 7 1 | 5040 | 10 |
| 10 | 6 2 | 48 | 10 |
| **Итого** | | | **100** |

**Возможный вариант решения задачи на языке Python:**

h, d = map(int, input().split())  
k = 0  
p = 1  
while (h > k \* d):  
 p \*= (h - k \* d)  
 k += 1  
print(p)

**Задача 3. Пик посещаемости**

**Максимальный балл: 100**

**Данные вводятся с клавиатуры или из файла input.txt, выводятся на экран или в файл output.txt.**

**Условие**

В торговом развлекательном комплексе (ТРК) в пик посетителей на кассах скапливалась большая очередь. Чтобы не было столпотворения администрация ТРК приняла решение в такие моменты увеличивать количество кассиров, для чего установили новые камеры слежения. Камеры регистрируют время прихода и ухода каждого посетителя. За один день камера получает N пар чисел, первое число – время прихода посетителя, второе – ухода. Помогите администрации ТРК определить какое максимальное число посетителей одновременно находятся в залах и промежуток времени, когда это происходит.

**Формат входных данных**

В первой строке вводится N – количество посетителей в день. Затем в каждой новой строке через пробел время прихода и ухода посетителей А В.

**Формат выходных данных**

Выведите в первой строке максимальное число посетителей, которые одновременно находились в ТРК, во второй строке через пробел начало и конец промежутка времени, когда это происходило.

**Ограничения**

1 ≤ N ≤ 100

1 ≤ А, В ≤ 86400 в секундах

**Пример входных и выходных данных**

|  |  |
| --- | --- |
| **Входные данные** | **Выходные данные** |
| 2  1 10  5 20 | 2  5 10 |
| 5  1 10  3 9  5 20  7 17  19 25 | 4  7 9 |

**Оценка задачи**

**Жюри 10 раз тестирует программу, один тест оценивается 10 баллами.**

Решение и критерии оценивания

**Пример входных и выходных данных для жюри**

|  |  |  |  |
| --- | --- | --- | --- |
| **№ теста** | **Входные данные** | **Выходные данные** | **Баллы** |
| 1 | 2  1 10  5 20 | 2  5 10 | 10 |
| 2 | 5  1 10  3 9  5 20  7 17  19 25 | 4  7 9 | 10 |
| 3 | 3  1 2  3 4  5 6 | 1  1 2 или 3 4 или 5 6 | 10 |
| 4 | 4  1 3  2 4  5 7  6 8 | 2  2 3 или 6 7 | 10 |
| 5 | 7  1 3  3 8  4 10  5 11  7 15  10 16  11 18 | 4  11 11 | 10 |
| 6 | 5  1 3  3 4  4 6  12 15  13 19 | 2  13 15 | 10 |
| 7 | 6  5 10  2 4  3 9  4 6  1 8  6 11 | 5  6 6 | 10 |
| 8 | 5  5 14  1 13  3 13  7 11  6 10 | 5  7 10 | 10 |
| 9 | 9  10 16  4 14  14 18  7 15  1 12  3 10  6 18  12 17  15 19 | 6  15 15 | 10 |
| 10 | 12  1 20  2 18  3 19  4 17  5 20  6 17  7 21  8 19  9 22  10 16  11 22  12 19 | 12  12 16 | 10 |
| **Итого** | | | **100** |

**Возможный вариант решения задачи на языке Pascal:**

program pik;

var

a,b: array[1..200] of real;

i, j, k, k0, n: integer;

x, y, c: real;

begin

readln(n);

for i:= 1 to n do

readln(a[i], b[i]);

for i:= 1 to n do

a[n+i]:= -b[i];

for i:= 1 to n+n-1 do

for j:= i+1 to n+n do

if abs(a[i]) > abs(a[j]) then

begin

c:= a[i];

a[i]:= a[j];

a[j]:= c;

end;

k:= 0;

k0:= 0;

for i:= 1 to n+n do

if a[i]>0 then

begin

k:= k + 1;

c:= a[i];

end else

begin

if k>=k0 then begin

k0:=k;

x:=c;

y:=-a[i];

end;

k:=k-1;

end;

writeln (k0);

writeln(x:4:0 , y:4:0 );

end.

**Задача 4. По алфавиту**

**Максимальный балл: 100**

**Данные вводятся с клавиатуры или из файла input.txt, выводятся на экран или в файл output.txt.**

**Условие**

Дана строка S, содержащая различные символы, цифры, прописные и строчные русские буквы длиной не более 250 символов. Если буквы в строке упорядочены по алфавиту, то вывести «ОК»; в противном случае вывести номер первого символа и сам символ, нарушающий порядок.

**Формат входных данных**

В единственной строке вводится набор из различных символов, букв, цифр.

**Формат выходных данных**

Вывести «ОК», если буквы в строке упорядочены по алфавиту, иначе номер первого символа и сам символ строки, нарушающего алфавитный порядок.

**Ограничения**

1 ≤ S ≤ 250

**Пример входных и выходных данных**

|  |  |
| --- | --- |
| **Входные данные** | **Выходные данные** |
| Аб+вклм1н | OK |
| 5A=в?бДЖ | 6 б |
| аВ3дЕ жи.ЛжмН7 | 11 ж |

**Оценка задачи**

**Жюри 10 раз тестирует программу, один тест оценивается 10 баллами.**

Решение и критерии оценивания

**Пример входных и выходных данных для жюри**

|  |  |  |  |
| --- | --- | --- | --- |
| **№ теста** | **Входные данные** | **Выходные данные** | **Баллы** |
| 1 | Аб+вклм1н | OK | 10 |
| 2 | 5A=в?бДЖ | 6 б | 10 |
| 3 | аВ3дЕ жи.ЛжмН7 | 11 ж | 10 |
| 4 | 123456789Аа1 | 11 а | 10 |
| 5 | 123кК | 5 К | 10 |
| 6 | Вг9ДёлМн6оПмрс | 12 м | 10 |
| 7 | Вгде1ёжЗ3ийК5лм7нО8пр0сТхцяшэюя | 28 ш | 10 |
| 8 | аАаА | 2 А | 10 |
| 9 | 123Эю34Я | ОК | 10 |
| 10 | (р8/сТ)т | 8 т | 10 |
| **Итого** | | | **100** |

**Возможное решение задачи на языке Pascal:**

**var** s: string;

c, b, b1: char;

i, n, j: integer;

**begin**

readln(s);

c := chr(ord('А')-1);

i := 1;

n := 1;

j := 0;

b1 := s[i];

**while** (i<= length(s)) **and** (n = 1) **do**

**begin**

b := s[i];

s[i] := upcase(s[i]);

**if** s[i] **in** ['А'..'Я'] **then begin**

j:=1;

**if** c < s[i] **then** c := s[i] **else begin** n := i; b1:=b; **end**; **end**;

inc(i)

**end**;

**if** (n=1) **and** (j = 1) **then** write('OK') **else** writeln (n, ' ', b1);

**end**.

**Задача 5. Термиты**

**Максимальный балл: 100**

**Данные вводятся с клавиатуры или из файла input.txt, выводятся на экран или в файл output.txt.**

**Условие**

На квадратный стол, поверхность которого разбита на клетки напали термиты. Если накапать из пипетки яд в некоторые клетки, то от убьет часть термитов: каждая капля яда убивает 8 термитов в той клетке, куда она попала, и по 4 термита в восьми (или менее, если клетка находится с краю поля) соседних клетках. Выясните, сколько термитов осталось.

**Формат входных данных**

На первой строке вводится натуральное число n — размер квадратного стола из клеток, на который напали термиты.

Далее следует n2 чисел на отдельных строках, которые описывают количество термитов в каждой клетке ряд за рядом сверху вниз, внутри ряда слева направо.

Далее следует натуральное число k — количество капель яд. Затем k пар чисел на отдельных строках — координаты клеток, в которые попали капли; сначала следует номер столбца (столбцы нумеруются слева направо), затем номер ряда (ряды нумеруются сверху вниз).

**Формат выходных данных**

Выводится количество термитов, выживших в каждой клетке.

**Ограничения**

Координаты отсчитываются с нуля, т. е. принимают значения от 0 до n-1, n≥3.

**Пример входных и выходных данных**

|  |  |
| --- | --- |
| **Входные данные** | **Выходные данные** |
| 3  10  10  9  10  10  10  10  10  10  2  1  1  0  0 | 0 2 5  2 0 6  6 6 6 |
| 3  5  5  5  5  5  5  5  5  5  1  1  0 | 1 0 1  1 1 1  5 5 5 |

**Оценка задачи**

**Жюри 10 раз тестирует программу, один тест оценивается 10 баллами.**

Решение и критерии оценивания

**Пример входных и выходных данных для жюри**

|  |  |  |  |
| --- | --- | --- | --- |
| **№ теста** | **Входные данные** | **Выходные данные** | **Баллы** |
| 1 | 3  10  10  9  10  10  10  10  10  10  2  1  1  0  0 | 0 2 5  2 0 6  6 6 6 | 10 |
| 2 | 3  5  5  5  5  5  5  5  5  5  1  1  0 | 1 0 1  1 1 1  5 5 5 | 10 |
| 3 | 4  4  6  8  2  4  7  12  15  16  42  17  25  34  21  19  10  4  2  1  1  0  0  3  3  2 | 0 0 0 0  0 0 0 7  12 34 9 13  26 17 15 6 | 10 |
| 4 | 4  4  6  18  2  4  7  2  15  16  2  17  25  34  21  9  10  4  0  1  2  0  1  3  3  2 | 0 0 10 0  0 0 0 7  8 0 9 17  30 13 1 6 | 10 |
| 5 | 4  14  6  18  12  14  7  12  15  16  29  17  25  24  25  9  11  3  1  1  2  2  3  2 | 10 2 14 12  10 0 0 7  12 21 1 13  24 21 1 3 | 10 |
| 6 | 4  14  26  18  12  24  27  12  15  16  29  17  25  24  25  29  21  2  1  2  2  3 | 14 26 18 12  20 23 8 15  12 17 9 21  20 17 17 17 | 10 |
| 7 | 3  4  26  1  12  4  27  2  1  16  5  0  0  2  1  1  1  2  2  2  0 | 0 10 0  4 0 7  0 0 0 | 10 |
| 8 | 4  12  6  15  4  18  9  12  11  15  10  19  7  14  3  6  11  7  0  2  1  1  3  2  0  2  1  3  1  2  2  2 | 8 2 11 4  2 0 0 3  0 0 0 0  0 0 0 3 | 10 |
| 9 | 3  42  49  52  31  47  62  14  27  30  1  2  1 | 42 45 48  31 43 54  14 23 26 | 10 |
| 10 | 4  2  6  5  4  8  9  12  1  15  10  9  7  14  3  6  11  3  1  1  0  2  1  3 | 0 2 1 4  0 0 8 1  0 0 1 7  6 0 2 11 | 10 |
| **Итого** | | | **100** |

**Возможное решение задачи на языке Python:**

n = int(input())

table = [[0] \* n for \_ in range(n)]

for i in range(n):

for j in range(n):

table[i][j] = int(input())

k = int(input())

strikes = [[0, 0] for \_ in range(k)]

for i in range(k):

for j in range(2):

strikes[i][j] = int(input())

n -= 1

for s in strikes:

x = s[0]

y = s[-1]

table[y][x] -= 8

if x != n:

table[y][x + 1] -= 4

if x != 0:

table[y][x - 1] -= 4

if y != n:

table[y + 1][x] -= 4

if y != 0:

table[y - 1][x] -= 4

if y != 0 and x != 0:

table[y - 1][x - 1] -= 4

if y != 0 and x != n:

table[y - 1][x + 1] -= 4

if y != n and x != n:

table[y + 1][x + 1] -= 4

if y != n and x != 0:

table[y + 1][x - 1] -= 4

for t in table:

for e in t:

if e >= 0:

print(e, end='\t')

else:

print(0, end='\t')

print()

**Задача 6. Шахматы**

**Максимальный балл: 100**

**Данные вводятся с клавиатуры или из файла input.txt, выводятся на экран или в файл output.txt.**

**Условие**

Александр Сергеевич с детства очень увлекался шахматами. К шестому классу он выигрывал у всех своих одноклассников, перечитал все книжки по шахматам, решил все задачи, в них приведенные. Долго не мог найти достойного противника. И стало ему скучно.

Но тут познакомился он с программированием. Сначала был он равнодушен к нему, но узнав, что можно написать программу для игры в шахматы, решил он создать сильнейшего противника. «И как же подступиться к этой задаче,» – думал он. – «А начну я с решения простых задач!».

Обрадовался Александр Сергеевич своей идее и приступил к решению. Взял первую простую задачу, увидел, что используются там три фигуры: ферзь, ладья, да конь. И требуется для выбора решения определить количество полей доски, которые не находятся под боем.

Задумался крепко наш герой, но не отчаялся. Помогите Александру Сергеевичу решить задачу. Будем считать, что фигуры могут бить через другие фигуры.

**Формат входных данных**

В строке прописаны через один пробел координаты расположенных трех фигур: ферзя, ладьи и коня соответственно. Каждая координата состоит из одной латинской буквы (от А до Н) и одной цифры (от 1 до 8).

**Формат выходных данных**

Вывести количество пустых полей, которые не бьют указанные во входных данных фигуры.

**Пояснения**

|  |  |  |
| --- | --- | --- |
| Конь | Ладья | Ферзь |
| Конь ходит по необычной траектории, напоминающей букву «Г» – передвигается на 2 клетки вперед и одну клетку в сторону. | Ладья ходит и бьет по вертикали и горизонтали на любое число клеток. | Ферзь ходит во все стороны по вертикали, горизонтали и диагонали на любое число клеток доски. |
| к | н | ф |

**Пример входных и выходных данных**

|  |  |
| --- | --- |
| **Входные данные** | **Выходные данные** |
| D1 D3 E5 | 32 |
| B1 A3 E5 | 25 |

**Оценка задачи**

**Жюри 10 раз тестирует программу, один тест оценивается 10 баллами.**

Решение и критерии оценивания

**Пример входных и выходных данных для жюри**

|  |  |  |  |
| --- | --- | --- | --- |
| **№ теста** | **Входные данные** | **Выходные данные** | **Баллы** |
| 1 | D1 D3 E5 | 32 | 10 |
| 2 | B1 A3 E5 | 25 | 10 |
| 3 | A8 G8 H1 | 35 | 10 |
| 4 | B2 C5 D4 | 28 | 10 |
| 5 | C3 G3 E7 | 29 | 10 |
| 6 | D6 F7 C2 | 25 | 10 |
| 7 | H8 E4 F3 | 27 | 10 |
| 8 | F7 D2 A6 | 25 | 10 |
| 9 | G5 F3 B5 | 26 | 10 |
| 10 | D3 A4 G4 | 23 | 10 |
| **Итого** | | | **100** |

**Возможный вариант решения задачи на языке Pascal:**

**const**n=8;

**var** a:**array**[1..n,1..n] **of** integer;

hf, gf, hl, gl, hk, gk, c, i, j, k :integer;

s:string;

**begin**

readln (s);

hf:=ord(s[1])-64;

val(s[2],gf,c);

hl:=ord(s[4])-64;

val(s[5],gl,c);

hk:=ord(s[7])-64;

val(s[8],gk,c);

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

**for** j:=1 **to** n **do**

a[i,j]:=0;

a[gf,hf]:=1;

a[gl,hl]:=1;

a[gk,hk]:=1;

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

**for** j:=1 **to** n **do**

**begin**

**if** (j=hf)**or**(i=gf) **or** (abs(j-hf)=abs(i-gf))**then** a[i,j]:=1;

**if**(j=hl)**or**(i=gl)**then** a[i,j]:=1;

**if**((abs(j-hk)=1)**and**(abs(i-gk)=2))**or**((abs(j-hk)=2)**and**(abs(i-gk)=1)) **then** a[i,j]:=1;

**end**;

k:=0;

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

**for** j:=1 **to** n **do**

**if** a[i,j]=0 **then** {begin} k:=k+1; {writeln(i, ' ', j) end;}

writeln;

write(k);

**end**.