def find\_min\_sum(filename):

with open(filename, 'r') as f:

n = int(f.readline())

sm = [float('inf')] \* 10 # Инициализация списка для минимальных сумм

sm[0] = 0 # Начальная сумма с остатком 0

for \_ in range(n):

x, y = map(int, f.readline().split())

sm2 = [float('inf')] \* 10 # Временный список для новых сумм

for j in range(10):

if sm[j] == float('inf'):

continue # Пропускаем, если сумма для этого остатка недостижима

# Пробуем добавить x и y к текущей сумме

r1 = sm[j] + x

r2 = sm[j] + y

# Обновляем sm2 для новых остатков

sm2[r1 % 10] = min(sm2[r1 % 10], r1)

sm2[r2 % 10] = min(sm2[r2 % 10], r2)

sm = sm2 # Обновляем основной список

return sm[6] # Возвращаем сумму с остатком 6

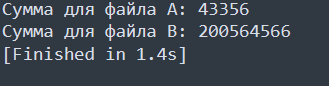
# Вызов функции для файлов A и B

sum\_a = find\_min\_sum('aX.txt')

sum\_b = find\_min\_sum('bX.txt')

print(f"Сумма для файла A: {sum\_a}")

print(f"Сумма для файла B: {sum\_b}")



Pascal

program MinSum;

const

MaxN = 100000;

MaxValue = 10000;

type

TArray = array[0..9] of Int64;

var

n, i: Integer;

x, y: Integer;

sm, sm2: TArray;

f: Text;

procedure InitializeArray(var arr: TArray);

var

i: Integer;

begin

for i := 0 to 9 do

arr[i] := High(Int64); // Инициализация большим числом

arr[0] := 0; // Начальная сумма с остатком 0

end;

function FindMinSum(filename: string): Int64;

begin

Assign(f, filename);

Reset(f);

ReadLn(f, n);

InitializeArray(sm);

for i := 1 to n do

begin

ReadLn(f, x, y);

InitializeArray(sm2);

for var j := 0 to 9 do

begin

if sm[j] = High(Int64) then

Continue; // Пропускаем недостижимые суммы

var r1 := sm[j] + x;

var r2 := sm[j] + y;

sm2[r1 mod 10] := Min(sm2[r1 mod 10], r1);

sm2[r2 mod 10] := Min(sm2[r2 mod 10], r2);

end;

sm := sm2; // Обновляем основной массив

end;

Close(f);

Result := sm[6]; // Возвращаем сумму с остатком 6

end;

begin

var sum\_a := FindMinSum('aX.txt');

var sum\_b := FindMinSum('bX.txt');

WriteLn('Сумма для файла A: ', sum\_a);

WriteLn('Сумма для файла B: ', sum\_b);

end.



