

Разбор задач

Задача 1. Расстановка знаков

Наибольший результат можно получить, если перемножить три наибольших числа и перед ними поставить знак $+$, то есть выражение будет заканчиваться на $+7*8*9$. Также в выражении будут два слагаемых, которые будут вычитаться. Их нужно сделать как можно меньше, поэтому используем операцию деления, при помощи которой можно получить выражения $-2/3$, $-3/4$, $-4/5$, $-5/6$. Из этих выражений самое маленькое по модулю — это $-2/3$, а следующее по величине, не содержащее цифры 2 и 3, это $-4/5$. Перед цифрой 6 поставим знак $+$.

Ответ: $1 - 2/3 - 4/5 + 6 + 7 * 8 * 9$.

Задача 2. Замена символов

В строке «АВААВВ» нужно букву А заменить на 1, а букву В заменить на 2, получится число 121122.

В строке «ВАВАСССА», чтобы получить минимальное число, нужно заменить В на 1, А на 2 и С на 3, получится 12123332.

В строке «DВАССАСDС» используются четыре цифры, поэтому придётся использовать двузначные числа. Чтобы результат получился как можно меньше, прежде всего нужно минимизировать количество цифр в результате. Поэтому для замен будем использовать числа 1, 2, 3 и 10. В данной строке буква В встречается один раз, остальные буквы встречаются два и более раза, поэтому число будет самым коротким, если В заменить на 10. Оставшиеся буквы заменим на 1, 2, 3 в порядке их появления в строке, то есть D на 1, А на 2, С на 3. Получится 110233213.

В строке «ЕЕЕЕЕDDDDDСССВВА» буквы А, В, С, D, Е встречаются 1, 2, 3, 4, 5 раз соответственно. Поэтому наиболее частые буквы С, D, Е заменим на однозначные числа 1, 2, 3, а буквы А и В — на двузначные числа 10 и 20. Какие именно буквы на какие цифры менять определяется тем, какая буква встречается в строке раньше. Ответ: 111112222333101020.

В строке «АВВССDDEEFFFF» буква А встречается один раз, она будет заменена на двузначное число. Буква F встречается три раза, она будет заменена на однозначное число. Буквы В, С, D, Е встречаются по два раза, и для них можно использовать как однозначные, так и двузначные числа. Пройдём по строке слева направо. А заменим на минимальное двузначное число 10. За ней идёт буква В, её можно заменить на двузначное 20 или однозначное 1. Чтобы результат был меньше, нужно использовать 1. Следующую букву С можно заменить на 20 или 2, во всех случаях длина результата будет одинаковой. Но лучше использовать число 20, т.к. в нём после цифры 2 идёт минимальная цифра 0. Для буквы D будем использовать число 3, а для следующей буквы Е — число 30, т.к. число 3 нужно оставить для буквы F. Ответ: 10112020223030333.

Задача 3. Фабрика

Для решения задачи нужна операция деления с округлением вверх, то есть если результатом деления является дробное число, его нужно округлить до большего целого числа. Например, если в одной коробке 5 пачек бумаги, то для того, чтобы заказать 27 пачек бумаги, необходимо 6 коробок, т.к. $27/5 = 5,4$. Будем обозначать результат деления a на b с округлением вверх, как $\lceil a/b \rceil$.

Для того, чтобы заказать t пачек бумаги в коробках по k штук необходимо $\lceil t/k \rceil$ коробок. Общее число пачек бумаги в этих коробках равно $p = k \cdot \lceil t/k \rceil$ коробок. Количество дней, за которое фабрика произведёт такое количество бумаги, производя ежедневно по n пачек, равно $\lceil p/n \rceil$.

Далее нам нужно вычислить эти значения для 20 заданных наборов n , k , t . Это можно сделать разными способами: используя электронные таблицы, написав программу или даже выполняя вычисления на калькуляторе (например, используя приложение «калькулятор» на компьютере).

В электронных таблицах можно вычислить частное при помощи операции «/», а затем результат округлить вверх при помощи функции ROUNDUP (в русском интерфейсе ОКРУГЛВВЕРХ). Запишем в ячейку D2 таблицы формулу $=\text{ROUNDUP}(\text{ROUNDUP}(C2/B2)*B2/A2)$, скопируем её и вставим в ячейки блока D2:D21. Затем выделим этот блок, скопируем и вставим в поле для ввода ответа.

В программе на языке Python вычислить $\lceil a/b \rceil$ можно при помощи выражения $(a + b - 1) // b$. Поэтому для данных значений n , k , t ответ можно найти при помощи программы

```
p = (t + k - 1) // k * k  
ans = (t1 + n - 1) // n
```

Далее можно написать программу, которая вычисляет ответ для введённых чисел n , k , t и ввести 20 тестовых примеров по одному, или написать одну программу, которая будет выдавать ответ сразу для всех 20 тестовых примеров.

Ответ на это задание.

```
2  
7  
106  
5  
175  
90  
38  
672  
301  
74  
1044  
133  
5664  
41464  
3441  
129283  
1218816  
4811850  
60638  
38536292
```

Задача 4. Большой квадрат

Рассмотрим разные способы разместить квадрат внутри двух прямоугольников. Если квадрат полностью размещён внутри первого прямоугольника, то максимально возможная длина стороны равна наименьшей стороне прямоугольника $\min(a, b)$. Если он полностью размещён внутри второго прямоугольника, то $\min(c, d)$. Наконец, рассмотрим вариант, когда для вырезания квадрата понадобятся оба прямоугольника, как на картинке в условии. Пусть прямоугольники сложены так, что стороны a и c являются продолжением друг друга, а стороны b и d касаются. Тогда длина стороны квадрата не может быть больше значений $a + c$, b и d , и ответом будет $\min(a + c, b, d)$. Очевидно, что в этом случае нужно выбрать стороны так, чтобы a была наименьшей стороной первого прямоугольника, то есть $a \leq b$. Аналогично, c должна быть наименьшей стороной второго прямоугольника, то есть $c \leq d$.

Итак, если упорядочить стороны прямоугольников, то есть сделать так, что $a \leq b$ и $c \leq d$, то ответ равен $\max(a, c, \min(a + c, b, d))$. Пример такого решения.

```
a = int(input())  
b = int(input())  
c = int(input())  
d = int(input())  
if a > b:  
    a, b = b, a  
if c > d:  
    c, d = d, c  
print(max(a, c, min(a + c, b, d)))
```

Задача 5. Перевозка грузов

Считаем число n , затем считаем циклом n масс, увеличивая на 1 одну из трёх переменных `cnt_heavy`, `cnt_medium`, `cnt_light` — количество тяжёлых, средних и лёгких грузов. После считывания данных найдём ответ. Тяжёлые грузы занимают по одной тележке, поэтому в ответе будет слагаемое `cnt_heavy`. Общее число лёгких и средних грузов равно `cnt_medium + cnt_light`. На каждую тележку можно поместить не более двух таких грузов, поэтому число тележек не может быть меньше половины от этой суммы, округлённой вверх. Также на каждую тележку можно положить не более одного среднего груза, поэтому число тележек для перевозки средних и лёгких грузов не может быть меньше, чем `cnt_medium`. Поэтому число тележек для перевозки лёгких и средних грузов равно $\max(\text{cnt_medium}, (\text{cnt_medium} + \text{cnt_light} + 1) // 2)$.

Пример решения на языке Python.

```
n = int(input())
cnt_heavy = 0
cnt_medium = 0
cnt_light = 0

for i in range(n):
    a = int(input())
    if a >= 100:
        cnt_heavy += 1
    elif a >= 50:
        cnt_medium += 1
    else:
        cnt_light += 1
print(cnt_heavy + max(cnt_medium, (cnt_medium + cnt_light + 1) // 2))
```

Задача 6. План эвакуации

Если комната имеет координаты (r, c) , то расстояние до лестницы с координатами (r_i, c_i) равно $|r - r_i| + |c - c_i|$ (так называемое «манхэттенское расстояние»). Посчитаем минимум расстояний от комнаты до двух лестниц. Из комнаты нужно перейти в ту из четырёх соседних комнат, для которой минимум расстояний до лестниц будет меньше, чем в этой комнате. Именно в эту комнату и направим стрелку из текущей комнаты. Задача имеет только реализационную трудность.

В примере решения ниже используются вспомогательные функции. `dist` возвращает расстояние между двумя комнатами, а `dist_to_exit` — расстояние от комнаты до ближайшей лестницы. Вложенными циклами проходим по всем комнатам. Для перебора направлений переходов в соседние комнаты удобно использовать цикл, в котором переменная `s` — это символ, соответствующий направлению перемещения, а переменные `dx` и `dy` — это значение изменения координат при переходе в данном направлении.

```
n = int(input())
m = int(input())
y1 = int(input())
x1 = int(input())
y2 = int(input())
x2 = int(input())

def dist(a1, b1, a2, b2):
    return abs(a1 - a2) + abs(b1 - b2)

def dist_to_exit(y, x):
    return min(dist(y, x, y1, x1), dist(y, x, y2, x2))

for y in range(1, n + 1):
```

```
for x in range(1, m + 1):
    curr_dist = dist_to_exit(y, x)
    move = 'S'
    for c, dy, dx in [('^', -1, 0), ('v', 1, 0), ('<', 0, -1), ('>', 0, 1)]:
        if dist_to_exit(y + dy, x + dx) < curr_dist:
            move = c
    print(move, end=' ')
print()
```

Многие участники написали решение по-другому. Для каждой комнаты сначала определим, к какому из двух выходов нужно двигаться, сравнив расстояния от комнаты до выходов. Пусть координаты рассматриваемой комнаты (r, c) , а ближайший выход находится в комнате (r_i, c_i) . Если $r < r_i$, нужно вывести указатель «вниз», если $r > r_i$ — указатель «вверх». Если $c < c_i$ — указатель «вправо», а если $c > c_i$ — указатель «влево».

Наконец, встречались и решения с нахождением кратчайшего маршрута при помощи алгоритма обхода графа в ширину, но в этой задаче это избыточно сложное решение.