Задача A. Самое частое слово

Закинем все слова в map. Теперь пройдем по мапу и найдем элемент с максимальным вторым значением.

```
for (auto it : mapp) {
    if (it.second > max_count) {
        max_count = it.second;
        ans = it.first;
    }
}
```

Задача B. Степени двойки

Чтобы решить эту задачу, нам нужно использовать map cnt и сохранить в нем, сколько раз каждое целое число встречается в данном массиве.

Затем нам нужно перебрать все заданные числа с переменной i. Пусть текущее число равно a_i. Для этого числа нам нужно перебрать все возможные степени числа 2 (количество этих степеней не более 31). Пусть текущая степень равна cur. Затем нам нужно сделать $cnt[a_i]$ — и добавить значение $cnt[a_i - cur]$ к ответу.

Задача C. A и B и ошибки компиляции

В этой задаче вам было дано 3 массива. Второй содержал все элементы первого, за исключением одного, третий массив содержал все элементы второго, за исключением одного. Необходимо было вывести удаленные элементы.

Решение: Я опишу простейшее решение этой задачи: Обозначим сумму элементов в первом, втором и третьем массиве как a, b и c соответственно. Ответом будут числа $a - b$ и $b - c$

Также существуют множество других решений, использующих map.

Задача D. Симметрическая разность множеств

Решение на Python совсем простое. В этом языке программирования есть операция симметрическая разность множеств.

```
ans = sorted(list(set(a^b)))
```

На C++ нужно будет приложить немного больше усилий. Закинем все элементы первого множества в set. Пойдем по второму и будем выкидывать если элемент встречается в обоих множествах. Ответом является все невыкинутые множества из первого и второго множества.

Задача E. Множество

Реализуйте три операции используя set. `+` — добавить в set. `-` — удалить из set. `?` — проверить наличие в set. Все эти функции были показаны на лекции.

Задача F. Словарь

Поддержим операции с помощью map

```
if (tp==1){
    string s;
    cin>>s;
    int k;
    cin>>k;
    z[s]+=k;
} else{
    string s;
    cin>>s;
    if (!z.count(s)){
        cout<<"ERROR\n";
```

```
    } else{
        cout<<z[s]<<"\n";
    }
}
```

Задача G. Игра в перерыве

Будем разбивать каждое число на два, пока можно — хуже от этого не будет, потом можно все собрать обратно. Теперь у нас есть большой массив, но хранить его полностью необязательно — в нем много чисел повторяются, поэтому будет хранить пары (x, cnt), где x — число, а cnt — сколько раз оно встречается. Например, если изначально у нас был массив [2, 2, 4, 6, 12], то хранить мы будем массив [(1, 8),(3, 5)]. Делать это можно например с помощью ассоциативных массивов, которые в языке C++ реализованы в контейнере std::map.

Теперь осталось заметить, что две различных пары нельзя соединять друг с другом, потому что числа в них не равны, а следовательно производить большие числа можно только внутри одной пары.

Несложно заметить, что максимальное число, которое можно получить из пары (x, cnt) равно $x \cdot 2^k$, где k — максимальное число, удовлетворяющее неравенству $2^k \leqslant cnt$. Осталось по всем парам взять максимум, это и будет ответом на задачу.

Задача H. Минимальный индекс

Задание аналогично предыдущему, храним map (значение, последнее вхождение). Когда нашлось значение, выводим разницу соответствующих индексов. Если за первые миллион значений не нашлось повторений, то ответ GREATER.

Задача I. Встречалось ли число раньше

Пройдем с set. Проверяем, что число есть в set. Выводим нужный ответ. После этого добавляем элемент в set.

Задача J. Количество различных чисел

Добавляем все элементы в set. В конце выводим размер set.