CSE 4404-Algorithms Lab. Winter 2022
Date: March 14, 2023.
Target Group: A
Topic: Bitmask DP

Instructions:
- Task naming format: fullID_L06_T01_1A.c/CPP
- **Solutions with less efficient approaches will be considered for partial marks.**

Task 01:

Given an integer array ***nums*** and an integer ***k***, return true if it is possible to divide this array into ***k*** non-empty subsets whose sums are all equal.

- Leetcode Problem Link: Partition to K Equal Sum Subsets

Sample Examples
Example 1: **Input:** nums = [4,3,2,3,5,2,1], k = 4 **Output:** true **Explanation:** It is possible to divide it into 4 subsets (5), (1, 4), (2,3), (2,3) with equal sums. **Example 2:** **Input:** nums = [1,2,3,4], k = 3 **Output:** false

```
class Solution {
public:
    int dp[1<<16];
    int allmask;
    int part_sum;
    vector<int>vec;

    bool isOn(int N,int pos)
    {
        return (bool)(N & (1<<pos)) ;
    }
    int turnOn(int N,int pos)
    {
        return (N | (1<<pos));
    }
```

```
    int Sum(int mask)
    {
        int cnt = 0;
        for(int i=0;i<16;i++)
        {
            if(isOn(mask,i)) cnt+=vec[i];
        }
        return cnt;
    }

    int call(int mask)
    {

        if(mask==allmask) return 1;
        if(dp[mask]!=-1) return dp[mask];

        int ret = 0;
        int setCnt = Sum(mask)/part_sum;
        for(int i=0;i<vec.size();i++)
        {
            if(isOn(mask,i)) continue;

            if(Sum(turnOn(mask,i)) <= (setCnt+1)*part_sum)
                ret =ret or call(turnOn(mask,i));

        }
        return dp[mask] = ret;
    }

    bool canPartitionKSubsets(vector<int>& nums, int k) {
        int cnt = 0;
        for(int i=0;i<nums.size();i++) cnt += nums[i];

        if(cnt % k) return false;

        part_sum = cnt / k;
        vec = nums;
        allmask = (1<<nums.size()) - 1;

        memset(dp,-1,sizeof(dp));

        if(call(0)) return true;
        return false;
    }
};
```

Task 02:

You work in a company which organizes marriages. Marriages are not that easy to fulfill, so the job is quite difficult for you. The job gets even more difficult when people have lots and lots of preferences. Some prioritize family background, some prioritize education, etc. Your company is still losing money and you want to help overcome this financial crisis by arranging as many marriages as possible. So, you collected ***N*** bio-data of men and ***N*** bio-data of women. After analyzing quite a lot you calculated the priority index of each pair of men and women.

Finally, you want to arrange ***N*** marriage ceremonies such that the total priority index is maximized. Remember that each man should be paired with a woman and only monogamous families should be formed.

Input

Each Input contains an integer N ($1 \le n \le 16$) denoting the number of men or women. Each of the next N lines will contain N integers each. The j^{th} integer in the i^{th} line denotes the priority index between the i^{th} man and j^{th} woman. All the integers will be positive and not greater than 10000.

Output

For each case, print the case number and the maximum possible priority index after all the marriages have been arranged.

Sample Input	**Sample Output**
3 1 2 3 6 5 4 8 1 2	Output: 16
2 1 5 2 1	Output: 7