Problem 01

You are given m strings of small letters, now concatenate the given strings.

The first line of the input will contain a single positive integer m , then the next m lines will contain a strings.

In the time of concatenation give a space after each word. See the sample output for more clarification.

Sample Input :

3

Programming

is

fun

Sample Output :

Programming is fun

Problem 02

You are given a string S of (0-1) , (a-z) and (A-Z) . Now tell that whether the string is a binary string or not

Note – A binary is contain only 0 and 1.

Print YES if the string is a binary string otherwise print NO.

Sample Input 1:

aB1Bsi1001sd

Sample Output 1:

NO

Sample Input 2:

0011110

Sample Output 2:

YES

Problem 03

You are given a string S of small letters , Now calculate the cost of the string and tell that whether the cost of the string is power of two or not.

In this problem cost means the sum of the alphabetic order of the given string. Alphabetic order means

a = 1 , b = 2 , c = 3 ……. z = 26

And, power of two is a number of the form 2^n (2 to the power n) where n is an integer.

Now,If the cost is power of two print YES with cost (in this format 2^n) otherwise print NO.

Sample Input 1:	**Sample Output 1:**
abc	NO
Sample Input 2:	**Sample Output 2:**
bbca	YES cost = 2^3

Problem 04

You are given two matrix of size NxM where N is the row number and M is the column number. Now first line of the input will contain the value of N and M , and the next two line will contain the two matrix, Now perform matrix multiplication operation.

Sample Input:

3 3

1 2 3

4 5 6

7 8 9

1 2 3

4 5 6

7 8 9

Sample Output:

30 36 42

66 81 96

102 126 150

Problem 05

There's a chessboard of size 8×8 . R rooks are placed on it and all others cells are empty. Now , if the row or column number of two rooks are same they will attack each other and will vanish from the chessboard. It is guaranteed that

If you randomly choose two or more rooks from the given input at most two rooks column or row number will be same.

Now you need to tell how many rooks will safe at the end along with their location?

The first line of the input will contain a single positive integer R, The next line will contain the position of the R rooks in this (x,y) format where x is the row number of the rook and y is the column number of the rook.

Note – Here row and column starts from 1.

Sample Input :

4

1 2

3 3

2 4

3 1

Explanation – Here (3,3) and (3,1) location rooks will attack each other.

Sample Output :

Safe rooks – 2

1 2

2 4

Problem 06

You are given a matrix of size NxN where N is the row and column number. Now first line of the input will contain the value of N , and the next line will contain the matrix.

Now print the matrix in the following way –

>> 1st row and 1st column and skip the common value of them

>> 2nd row and 2nd column and skip the common value of them

>> nth row and nth column and skip the common value of them

See the sample output for more clarification

Sample Input:

3

1 2 3

4 5 6

7 8 9

Sample Output:

row-1 and column-1 = 2 3 4 7

row-2 and column-2 = 4 6 2 8

row-3 and column-3 = 7 8 3 6

Codeforces Problem Link –

1) https://codeforces.com/problemset/problem/1703/C

2) https://codeforces.com/problemset/problem/1454/A

3) https://codeforces.com/problemset/problem/1708/A

```c
// Problem 01
#include <stdio.h>
int main(){
   int m, i;
   scanf("%d", &m);

   char str[100], sen[100];
   for (i=0; i<m; i++){
      scanf("%s", str);
      if (i==0){
         strcpy(sen, str); // str copies to sen
      }
      else{
         strcat(strcat(sen, " "), str);
      }
   }
   puts(strcat(sen, " "));
   // puts(sen);
   return 0;
}
```

```c
// Problem 02
#include<stdio.h>
int main(){
   char s[100];
   scanf("%s", s);
   int i=0, flag=0;

   for(i=0; s[i]!='\0'; i++){
      if(!(s[i] == '0' || s[i]== '1')){
         flag=1;
         break;
      }
   }
   if(flag==0){
      printf("YES\n");
   }
   else{
      printf("NO\n");
   }
   return 0;
}
```

```c
// Problem 03
#include<stdio.h>
int main(){
   char str[1000];
   fgets(str, sizeof(str), stdin);
   int len = strlen(str)-1, sum=0, i;
   for(i=0; i<len; i++){
      sum+= str[i]-96;
   }
   for(i=0; i<=len; i++){
      if(pow(2, i)==sum){
         printf("Yes\nconst = 2^%d", i);
         return 0;
      }
      else if(pow(2, i)>sum){
         printf("NO\n");
         return 0;
      }
   }
   return 0;
}
```

```c
// Problem 04
#include<stdio.h>
int main(){
   int mat1[100][100], mat2[100][100],
mulMat[100][100];
   int row, column, value, i, j, k;
   scanf("%d %d", &row, &column);
   for(i=1; i<=row; i++){
      for(j=1; j<=column; j++){
         scanf("%d", &mat1[i][j]);
      }
   }
   for(i=1; i<=row; i++){
      for(j=1; j<=column; j++){
         scanf("%d", &mat2[i][j]);
      }
   }
   for(k=1; k<=row; k++){
      for(i=1; i<=row; i++){
         value =0;
         for(j=1; j<=column; j++){
            value = value + (mat1[k][j] *
mat2[j][i]);
         }
         mulMat[k][i] = value;
      }
   }
```

```
    for(i=1; i<=row; i++){
        for(j=1; j<=column; j++){
            printf("%d ",mulMat[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```

```
// Problem 05
#include<stdio.h>
int main(){
    int r=8,c=8;
    int chess[r][c];
    for(int i=1; i<=r; i++){
        for(int j=1; j<=c; j++){
            chess[i][j]=0;
        }
    }
    int n;
    scanf("%d", &n);
    while(n--) // at 0 loop becomes false
    {
        int row, col;
        scanf("%d %d", &row, &col);
        chess[row][col]=1;
    }
    for(int i=1; i<=r; i++){
        int count=0, j;
        for(int j=1; j<=c; j++){
            if(chess[i][j]==1)
                count++;
        }
        if(count>1){
            for(int j=1; j<=c; j++){
                chess[i][j]=0;
            }
        }
    }
    for(int j=1;j<=c; j++){
        int col_count = 0;
        for(int i=1; i<=r; i++){
            if(chess[j][i]==1){
                col_count++;
            }
        }
        if(col_count >1){
            for(int i=1; i<=r; i++){
                chess[j][i]=0;
            }
        }
    }
    printf("\n");
    for(int i=1; i<=r; i++){
        for(int j=1; j<=c; j++){
            if(chess[i][j]!=0){
                printf("%d %d\n", i, j);
            }
        }
    }
    return 0;
}
```

```
// Problem 06
#include<stdio.h>
int main(){
    int n;
    scanf("%d", &n);
    int arr[n][n];
    for(int i=1; i<=n; i++){
        for(int j=1; j<=n; j++){
            scanf("%d", &arr[i][j]);
        }
    }
    for(int i=1; i<=n; i++){
        printf("row-%d and column-%d =  ", i, i);
        for(int j=1; j<=n; j++){
            if(i!=j){
                printf("%d ", arr[i][j]);
            }
        }
        for(int k=1; k<=n; k++){
            if(i!=k){
                printf("%d ", arr[k][i]);
            }
        }
        printf("\n");
    }
}
```

```
// Cypher: Codeforces
#include <stdio.h>
int main(){
   int test_case, initial_total_number,
initial_position[105], number_of_move;
   char str[100];

   scanf("%d", &test_case);
   for(int k=0; k<test_case; k++){
      scanf("%d", &initial_total_number);
      for(int i=0; i<initial_total_number; i++){
         scanf("%d", &initial_position[i]);
      }
      for(int i=0; i<initial_total_number; i++){
         scanf("%d", &number_of_move);
         scanf("%s", str);
         for(int j=0; j<number_of_move; j++){
            if(str[j] == 'D' && initial_position[i] ==
9){
               initial_position[i] = 0;
            }
            else if(str[j] == 'U' &&
initial_position[i] == 0){
               initial_position[i] = 9;
            }
            else if(str[j] == 'D'){
               initial_position[i] +=1;
            }
            else if(str[j] == 'U'){
               initial_position[i] -=1;
            }
         }
      }
      for(int i=0; i<initial_total_number; i++)
         printf("%d\n", initial_position[i]);
   }
   return 0;
}
```

```
// Special Permutation: Codeforces
#include <stdio.h>
int main(){
   int test_case, total_index, arr[105];

   scanf("%d", &test_case);
   for(int j=0; j<test_case; j++){
      scanf("%d", &total_index);
      for(int i=0; i<total_index-1; i++){
         arr[i] = 2+i;
      }
      arr[total_index-1] = 1; // by deafult set
      for(int i=0; i<total_index; i++){
         printf("%d ", arr[i]);
      }
      printf("\n");
   }
   return 0;
}
```

```
// Difference Operations: Codeforces
#include<stdio.h>
int main(){
   int t, n, a[1000], i;
   scanf("%d", &t);
   while(t--){
      int sum=0;
      scanf("%d", &n);
      for(i=0; i<n; i++){
         scanf("%d", &a[i]);
         if(a[i]%a[0]==0)
            sum++;
      }
      if(sum==n)
         printf("YES\n");
      else
         printf("NO\n");
   }
   return 0;
}
```