

The preference order of courses for each testcase has been attached to the repository as a input files. The remaining i/p file has been attached with the o/p file.

Test Case1(profList)

Input:

```
x1_Profs
Prof1 Prof2

x_2Profs
Prof3 Prof4

x3_Profs
Prof5

FD_CDCS
CDC1
HD_CDCS
cdc1
FD_ELE
ELE1
HD_ELE
ele1
```

Output:

```
//definite_pairs
CDC1: Prof4 Prof4
cdc1: Prof5 Prof5

//combination1 with respect to definite_pairs
ELE1: Prof1 Prof2
ele1: Prof5 Prof3

//combination2 with respect to definite_pairs
ele1: Prof3 Prof3
ELE1: Prof1 Prof3

//combination3 with respect to definite_pairs
ele1: Prof2 Prof5
ELE1: Prof2 Prof3

//combination4 with respect to definite_pairs
ele1: Prof1 Prof5
```

```

ELE1: Prof3 Prof3

//combination5 with respect to definite_pairs
ele1: Prof1 Prof2

```

The code provides with all possible combinations for a given set of preference order. This is the most basic test case with only 5 professors and 4 courses. Since the code uses recursion in multiple places, the output is also presented in a similar fashion instead of a systematic result. Initial CDCS are allotted according to the code that have only one possible match-up and they have been set under the category of definite_pairs. With respect to those definite pairs, remaining courses (here electives) are hence provided with all other possible matchups keeping in mind the definite_pairs, thereby providing all possible solutions.

Testcase 2(profList2)

input:

```

Prof1 Prof2 Prof3 Prof4 Prof5 Prof6

x_2Profs
Prof7 Prof8 Prof9

x3_Profs
Prof10 Prof11

FD_CDCS
CDC1 CDC2 CDC3 CDC4

HD_CDCS
HDCDC1

FD_ELE
ELE1 ELE2

HD_ELE
HDELE1 HDELE2

```

output:

```

//definite_pairs
CDC2: Prof1 Prof4
CDC3: Prof2 Prof6

//combinationA with respect to definite_pairs
CDC4: Prof3 Prof8

```

```
//combination1 with respect to combinationA
```

```
HDCDC1: Prof5 Prof9
```

```
//combination1.1
```

```
CDC1: Prof8 Prof10
```

```
//combination1.11
```

```
HDELE1: Prof9 Prof7
```

```
HDELE2: Prof7 Prof10
```

```
ELE1: Prof10 Prof11
```

```
ELE2: Prof11 Prof11
```

```
//combination1.12
```

```
ELE1: Prof11 Prof11
```

```
ELE2: Prof10 Prof11
```

```
HDELE2: Prof10 Prof10
```

```
//combination1.13
```

```
ELE1: Prof7 Prof11
```

```
ELE2: Prof11 Prof11
```

```
//combination1.14
```

```
ELE1: Prof11 Prof11
```

```
ELE2: Prof7 Prof11
```

```
HDELE1: Prof7 Prof7
```

```
HDELE2: Prof9 Prof10
```

```
//combination1.15
```

```
ELE1: Prof10 Prof11
```

```
ELE2: Prof11 Prof11
```

```
//combination1.16
```

```
ELE1: Prof11 Prof11
```

```
ELE2: Prof10 Prof11
```

```
HDELE2: Prof10 Prof10
```

```
//combination1.17
```

```
ELE1: Prof9 Prof11
```

```
ELE2: Prof11 Prof11
```

```
//combination1.18
```

```
ELE1: Prof11 Prof11
```

```
ELE2: Prof9 Prof11
```

```
//combination2.1 with respect to combinationA
```

```
CDC1: Prof10 Prof10
```

```
//combination2.11
```

```
HDELE1: Prof8 Prof9
```

```
HDELE2: Prof10 Prof7
```

```
ELE1: Prof7 Prof11
```

```
ELE2: Prof11 Prof11
```

```
//combination2.12
```

```
ELE1: Prof11 Prof11
```

```
ELE2: Prof7 Prof11
```

```
HDELE2: Prof7 Prof7
```

```
//combination2.13
```

```
ELE1: Prof10 Prof11
```

```
ELE2: Prof11 Prof11
```

```
//combination2.14
```

```
ELE1: Prof11 Prof11
```

```
ELE2: Prof10 Prof11
```

```
//combination3 with respect to combinationA
```

```
HDCDC1: Prof9 Prof9
```

```
//combination3.1
```

```
CDC1: Prof5 Prof10
```

```
//combination3.11
```

```
HDELE1: Prof8 Prof7
```

```
HDELE2: Prof7 Prof10
```

```
ELE1: Prof10 Prof11
```

```
ELE2: Prof11 Prof11
```

```
//combination3.12
```

```
ELE1: Prof11 Prof11
```

```
ELE2: Prof10 Prof11
```

```
HDELE2: Prof10 Prof10
```

```
//combination3.13
```

```
ELE1: Prof7 Prof11
```

```
ELE2: Prof11 Prof11

//combination3.14
ELE1: Prof11 Prof11
ELE2: Prof7 Prof11
HDELE1: Prof7 Prof7
HDELE2: Prof8 Prof10

//combination3.15
ELE1: Prof10 Prof11
ELE2: Prof11 Prof11

//combination3.16
ELE1: Prof11 Prof11
ELE2: Prof10 Prof11
HDELE2: Prof10 Prof10

//combination3.17
ELE1: Prof8 Prof11
ELE2: Prof11 Prof11

//combination3.18
ELE1: Prof11 Prof11
ELE2: Prof8 Prof11

//combination4 with respect to combinationA
HDCDC1: Prof9 Prof9

//combination4.1
CDC1: Prof10 Prof10

//combination4.11
HDELE1: Prof5 Prof8
HDELE2: Prof10 Prof7
ELE1: Prof7 Prof11
ELE2: Prof11 Prof11

//combination4.12
ELE1: Prof11 Prof11
ELE2: Prof7 Prof11
HDELE2: Prof7 Prof7

//combination4.13
```

```
ELE1: Prof10 Prof11  
ELE2: Prof11 Prof11
```

```
//combination4.14
```

```
ELE1: Prof11 Prof11  
ELE2: Prof10 Prof11
```

```
//combinationB with respect to definite_pairs  
CDC4: Prof8 Prof8
```

```
//combination5 with respect to combinationB  
HDCDC1: Prof3 Prof9
```

```
//combination5.1  
CDC1: Prof5 Prof10
```

```
//combination5.11  
HDELE1: Prof9 Prof7  
HDELE2: Prof7 Prof10  
ELE1: Prof10 Prof11  
ELE2: Prof11 Prof11
```

```
//combination5.12  
ELE1: Prof11 Prof11  
ELE2: Prof10 Prof11  
HDELE2: Prof10 Prof10
```

```
//combination5.13  
ELE1: Prof7 Prof11  
ELE2: Prof11 Prof11
```

```
//combination5.14  
ELE1: Prof11 Prof11
```

```
ELE2: Prof7 Prof11
HDELE1: Prof7 Prof7
HDELE2: Prof9 Prof10

//combination5.15
ELE1: Prof10 Prof11
ELE2: Prof11 Prof11

//combination5.16
ELE1: Prof11 Prof11
ELE2: Prof10 Prof11
HDELE2: Prof10 Prof10

//combination5.17
ELE1: Prof9 Prof11
ELE2: Prof11 Prof11

//combination5.18
ELE1: Prof11 Prof11
ELE2: Prof9 Prof11


//combination5.2 with respect to combinationB
CDC1: Prof10 Prof10
HDELE1: Prof5 Prof9
HDELE2: Prof10 Prof7
ELE1: Prof7 Prof11
ELE2: Prof11 Prof11

//combination5.21
ELE1: Prof11 Prof11
ELE2: Prof7 Prof11
HDELE2: Prof7 Prof7

//combination5.22
ELE1: Prof10 Prof11
ELE2: Prof11 Prof11

//combination5.23
```

```
ELE1: Prof11 Prof11
ELE2: Prof10 Prof11

//combination6 with respect to combinationB
HDCDC1: Prof9 Prof9

//combination6.1
CDC1: Prof3 Prof10

//combination6.11
HDELE1: Prof5 Prof7
HDELE2: Prof7 Prof10
ELE1: Prof10 Prof11
ELE2: Prof11 Prof11

//combination6.12
ELE1: Prof11 Prof11
ELE2: Prof10 Prof11
HDELE2: Prof10 Prof10

//combination6.13
ELE1: Prof7 Prof11
ELE2: Prof11 Prof11

//combination6.14
ELE1: Prof11 Prof11
ELE2: Prof7 Prof11
HDELE1: Prof7 Prof7
HDELE2: Prof5 Prof10

//combination6.15
ELE1: Prof10 Prof11
ELE2: Prof11 Prof11

//combination6.16
ELE1: Prof11 Prof11
ELE2: Prof10 Prof11
HDELE2: Prof10 Prof10
```



```

//combination6.17
ELE1: Prof5 Prof11
ELE2: Prof11 Prof11

//combination6.18
ELE1: Prof11 Prof11
ELE2: Prof5 Prof11

//combination6.2
CDC1: Prof10 Prof10

//combination6.21
HDELE1: Prof3 Prof5
HDELE2: Prof10 Prof7
ELE1: Prof7 Prof11
ELE2: Prof11 Prof11

//combination6.22
ELE1: Prof11 Prof11
ELE2: Prof7 Prof11
HDELE2: Prof7 Prof7

//combination6.23
ELE1: Prof10 Prof11
ELE2: Prof11 Prof11

//combination6.24
ELE1: Prof11 Prof11
ELE2: Prof10 Prof11

```

This testcase takes more number of professors and courses as inputs. This testcase has been taken to demonstrate that all possible outcomes are produced without the code crashing. There are these many outputs because the preference order for various professors has been set to clash to produce more possible combinations. The output first produces the definite_pairs, here CDC2 and CDC3; then with respect to the definite_pairs, CDC4 is allotted according to combinationA. With respect to combinationA the HDCDC is allotted marking Combination1, then CDC1 is allotted keeping in mind combination1, thereby finishing CDCs(combination 1.1). After that the electives are allotted in one specific permutation(combination 1.1.1). This is called recursively in the code so electives keep creating combinations possible with respect to combination1.1 . After this iteration, earlier recursive calls are completed forming combination2

and so on. Further going back, CombinationB is generated that flows in the same manner as CombinationA.

To summarise, the code generates a combination and recursively runs to generate more combinations for the earlier recursive call. Since the actual output format would be difficult to comprehend, comments have been added manually. However all possible combinations have been successfully generated.

Testcase 3(profList1)

input:

```
x1_Profs
Prof1 Prof2 Prof3 Prof4 Prof5 Prof6 Prof7 Prof8 Prof9 Prof10 Prof11 Prof12

x2_Profs
Prof13 Prof14 Prof15 Prof16 Prof17 Prof18

x3_Profs
Prof19 Prof20 Prof21 Prof22

FD_CDCS
CDC1 CDC2 CDC3 CDC4 CDC5 CDC6 CDC7 CDC8
HD_CDCS
HD_CDC1 HD_CDC2 HD_CDC3
FD_ELE
ELE1 ELE2 ELE3 ELE4 ELE5
HD_ELE
HD_ELE1 HD_ELE2 HD_ELE3
```

output:

```
//definite_pairs
HD_CDC2: Prof2 Prof3
HD_CDC3: Prof4 Prof5
HD_CDC1: Prof6 Prof9
CDC1: Prof12 Prof13
CDC2: Prof13 Prof14
CDC3: Prof14 Prof15
CDC4: Prof15 Prof16
CDC5: Prof16 Prof17
CDC6: Prof17 Prof18
CDC7: Prof18 Prof19
CDC8: Prof19 Prof19
ELE5: Prof1 Prof8
HD_ELE1: Prof7 Prof11
HD_ELE2: Prof10 Prof21
HD_ELE3: Prof21 Prof21
ELE3: Prof22 Prof22
```

```
ELE1: Prof20 Prof20
HD_ELE3: Prof21 Prof22
ELE4: Prof22 Prof22

//combination1 with respect to definite_pairs
ELE3: Prof20 Prof20
HD_ELE3: Prof22 Prof22
ELE4: Prof21 Prof21

//combination2 with respect to definite_pairs
ELE3: Prof20 Prof20
HD_ELE2: Prof10 Prof22
HD_ELE3: Prof22 Prof22

//combination3 with respect to definite_pairs
ELE3: Prof21 Prof21
ELE4: Prof20 Prof20
HD_ELE3: Prof22 Prof21

//combination4 with respect to definite_pairs
ELE4: Prof21 Prof21
ELE3: Prof20 Prof20
HD_ELE3: Prof21 Prof21

//combination5 with respect to definite_pairs
ELE4: Prof22 Prof22
ELE3: Prof20 Prof20
HD_ELE2: Prof21 Prof21
HD_ELE3: Prof10 Prof20

//combination6 with respect to definite_pairs
ELE4: Prof20 Prof20
ELE3: Prof22 Prof22
HD_ELE3: Prof10 Prof22

//combination7.1 with respect to definite_pairs
ELE4: Prof22 Prof22
ELE3: Prof20 Prof20
HD_ELE3: Prof20 Prof20
//combination7.2
HD_ELE3: Prof20 Prof22
//combination7.3
HD_ELE3: Prof22 Prof22
```

```

//combination8 with respect to definite_pairs
HD_ELE2: Prof21 Prof22
HD_ELE3: Prof10 Prof22
ELE4: Prof21 Prof21
ELE3: Prof20 Prof20

//combination9 with respect to definite_pairs
HD_ELE3: Prof10 Prof20
ELE3: Prof22 Prof22
ELE4: Prof20 Prof20
ELE2: Prof21 Prof21

```

This is the most relevant testcase with 22 professors and 19 courses. This testcase has been set realistically indicating preferences of professors in a manner so that not many of them clash. 0 has also been set as preference to indicate that a professor is not interested in teaching the course. Therefore, even though the input is larger than testcase2, the output combinations are lesser. The code works for a large number of inputs and is capable of producing a large number of outputs. The code has not crashed due to stack overflow in various attempts at testcases.

Testcase 4(profListERROR)

input:

```

x1_Profs
Prof1 Prof2

x_2Profs
Prof3 Prof4

x3_Profs
Prof5

FD_CDCS
CDC1
HD_CDCS
cdc1

FD_ELE
ELE1
HD_ELE
ele1

```

This tescase produces error as as no professor is allowed to fill 0 as preference in all CDCs. The error has been handled through try block.