

945 Loading a Cargo Ship

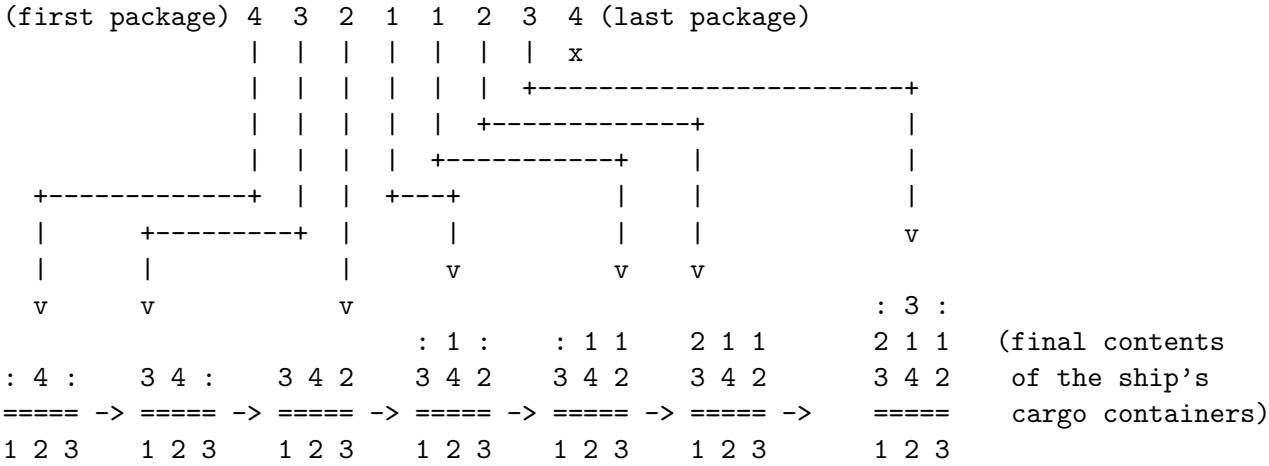
Ahoy there, software sailors! Your ship is in need of repairs! Last night a big wave swallowed the cargo routing computer (CRC) which is essential for logistic operations. Fortunately there is a replacement computer in the lower deck; however, there is no backup of the cargo routing software! Your mission is to reprogram the CRC assuming the following ship characteristics:

- The ship has up to 9 cargo containers (of different sizes) available for storing packages;
- The containers are numbered 1 to 9;
- Each cargo container has a specific maximum weight capacity, up to 999 tons;
- Each package weights at most 9 tons;
- The ship can only be loaded with no more than 999 packages (for good luck, the crew says).

The packages are loaded into the ship via a conveyor belt which is connected to a cargo router (controlled by the CRC). The cargo router then forwards each package to one of the available cargo containers, according to the following algorithm:

1. First, select only the cargo containers that carry the lowest number of packages;
2. Then, reduce the selection by choosing only the containers that have the greatest available weight capacity;
3. Further reduce the selection by picking the container with the lowest number;
4. If the selected cargo container cannot carry the package then the ship's loading ends.

To illustrate the cargo routing algorithm, consider a ship with 3 cargo containers (numbered 1, 2, and 3), respectively with 5, 10, and 5 tons of maximum weight capacity. Assume now there are 8 packages to be loaded, in sequence, into the ship, weighting 4, 3, 2, 1, 1, 2, 3, and 4 tons. In this scenario, the cargo routing algorithm should produce the following activity:



Note: the colon symbol ‘:’ is used to avoid presentation errors; see also the sample output

When the first package arrives (weighting 4 tons) all containers are empty, so rule 1 of the algorithm selects containers 1, 2, and 3. In step 2, because container 2 has the greatest available capacity (10 maximum tons - 0 used tons = 10 available tons), only container 2 remains selected. Rule 3 again selects container 2. In rule 4 the package is loaded into container 2 because its available capacity (10 tons) is greater than the 4 tons of the package.

The second package ends up in container 1: in rule 1, containers 1 and 3 are selected because both have the lowest number of packages (zero, in this case); rule 2 still selects the same containers since they have the same 5 tons of available weight capacity; rule 3 reduces the selection to just container 1; finally, rule 4 loads the package, weighting 3 tons, into container 1.

And so on, until all packages have been loaded into the ship, or, as in this scenario, until no cargo container can carry the current package, which prematurely ends the ship’s loading (check the situation of the last package, weighting 4 tons).

Hurry! (and remember to backup the cargo routing software)

Input

Input contains several test cases, separated by a blank line. Each test case describes the cargo containers and the packages to be loaded into the ship:

- The first line indicates the number c of cargo containers in the ship ($1 \leq c \leq 9$);
- The following c lines describe the maximum weight cw_i each container can carry ($1 \leq cw_i \leq 999$);
- Then comes an empty line;
- The next line indicates the number p of packages to be loaded into the ship ($1 \leq p \leq 999$);
- The following p lines describe the weight pw_i of each package ($1 \leq pw_i \leq 9$).

It is guaranteed that the weight of all packages is always lower or equal to the ship’s maximum weight capacity ($sum(pw_i) \leq sum(cw_i)$).

Output

The output for each test case displays the final contents of the ship’s cargo containers and presents a summary of total weights:

- The first lines represent the final contents of the cargo containers;
- Then comes an empty line;
- The next line displays the weight of all packages loaded into the ship;
- The following line shows the total unused weight the ship would still be able to carry;
- The last line indicates the total weight of the packages that could not be loaded into the ship.

Examine carefully the sample output to prevent presentation errors. Print a blank line between datasets.

Sample Input

```
3
5
10
5

8
4
3
2
1
1
2
3
4
```

Sample Output

```
: 3 :
2 1 1
3 4 2
=====
1 2 3
```

```
cargo weight: 16
unused weight: 4
unloaded weight: 4
```