Knapsack

HackerRank

Given an array of integers and a target sum, determine the sum nearest to but not exceeding the target that can be created. To create the sum, use any element of your array zero or more times.

For example, if $arr = [2, 3, 4]$ and your target sum is 10, you might select $[2, 2, 2, 2, 2]$, $[2, 2, 3, 3]$ or $[3, 3, 3, 1]$. In this case, you can arrive at exactly the target.

Function Description

Complete the *unboundedKnapsack* function in the editor below. It must return an integer that represents the sum nearest to without exceeding the target value.

unboundedKnapsack has the following parameter(s):

- *k*: an integer
- *arr*: an array of integers

Input Format

The first line contains an integer t, the number of test cases.

Each of the next t pairs of lines are as follows:
- The first line contains two integers n and k, the length of arr and the target sum.
- The second line contains n space separated integers $arr[i]$.

Constraints

$1 \leq t \leq 10$
$1 \leq n, k, arr[i] \leq 2000$

Output Format

Print the maximum sum for each test case which is as near as possible, but not exceeding, to the target sum on a separate line.

Sample Input

```
2
3 12
1 6 9
5 9
3 4 4 4 8
```

Sample Output

```
12
9
```

Explanation

In the first test case, one can pick {6, 6}. In the second, we can pick {3,3,3}.