

You have been given a vector/list **ARR** consisting of 'N' integers. You are also given a positive integer 'K'.

Let's define a vector/list **CONCAT** of size  $N * K$  formed by concatenating **ARR** 'K' times. For example, if **ARR** = [0, -1, 2] and  $K = 3$ , then **CONCAT** is given by [0, -1, 2, 0, -1, 2, 0, -1, 2].

Your task is to find the maximum possible sum of any non-empty subarray (contagious) of **CONCAT**.

#### Input Format:

The first line of input contains an integer 'T' representing the number of test cases or queries to be processed. Then the test case follows.

The first line of each test case contains two single space-separated integers 'N' and 'K' representing the size of the vector/list and the given integer, respectively.

The second line of each test case contains 'N' single space-separated integers representing the vector elements.

#### Output Format :

For each test case, print the maximum possible subarray sum of **CONCAT**.

Print the output of each test case in a separate line.

Note :

You do not need to print anything; it has already been taken care of. Just implement the function.

#### Constraints:

```
1 <= T <= 100
1 <= N <= 5*10^3
1 <= K <= 5*10^3
-10^5 <= ARR[i] <= 10^5
```

Where 'N' is the number of elements in the vector/list.

Time Limit: 1sec

#### Sample Input 1 :

```
2
2 3
1 3
3 2
1 -2 1
```

**Sample Output 1 :**

```
12
2
```

**Sample Output 1 Explanation:**

For the first test case, vector CONCAT is obtained by concatenating vector [1, 3] three times.

CONCAT = [1, 3, 1, 3, 1, 3]

The subarray with a maximum sum of 12 is [1, 3, 1, 3, 1, 3].

For the second test case, vector CONCAT is obtained by concatenating vector [1, -2, 1] two times.

CONCAT = [1, -2, 1, 1, -2, 1]

The subarray with a maximum sum of 2 is [1, 1].

**Sample Input 2 :**

```
1
2 3
-2 1
```

**Sample Output 2 :**

```
1
```