**Best Reward**

**Time Limit: 2000/1000 MS (Java/Others)    Memory Limit: 65536/65536 K (Java/Others)**

**Problem Description**

After an uphill battle, General Li won a great victory. Now the head of state decide to reward him with honor and treasures for his great exploit.   
  
One of these treasures is a necklace made up of 26 different kinds of gemstones, and the length of the necklace is n. (That is to say: n gemstones are stringed together to constitute this necklace, and each of these gemstones belongs to only one of the 26 kinds.)   
  
In accordance with the classical view, a necklace is valuable if and only if it is a palindrome - the necklace looks the same in either direction. However, the necklace we mentioned above may not a palindrome at the beginning. So the head of state decide to cut the necklace into two part, and then give both of them to General Li.   
  
All gemstones of the same kind has the same value (may be positive or negative because of their quality - some kinds are beautiful while some others may looks just like normal stones). A necklace that is palindrom has value equal to the sum of its gemstones' value. while a necklace that is not palindrom has value zero.   
  
Now the problem is: how to cut the given necklace so that the sum of the two necklaces's value is greatest. Output this value.

**Input**

The first line of input is a single integer T (1 ≤ T ≤ 10) - the number of test cases. The description of these test cases follows.   
  
For each test case, the first line is 26 integers: v1, v2, ..., v26 (-100 ≤ vi ≤ 100, 1 ≤ i ≤ 26), represent the value of gemstones of each kind.   
  
The second line of each test case is a string made up of charactor 'a' to 'z'. representing the necklace. Different charactor representing different kinds of gemstones, and the value of 'a' is v1, the value of 'b' is v2, ..., and so on. The length of the string is no more than 500000.

**Output**

Output a single Integer: the maximum value General Li can get from the necklace.

**Sample Input**

2

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

aba

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

acacac

**Sample Output**

1

6

**Source**

[2010 ACM-ICPC Multi-University Training Contest（18）——Host by TJU](http://acm.hdu.edu.cn/search.php?field=problem&key=2010+ACM-ICPC+Multi-University+Training+Contest%A3%A818%A3%A9%A1%AA%A1%AAHost+by+TJU+&source=1&searchmode=source)