

N gangsters are going to a restaurant. The i -th gangster comes at the time T_i and has the *prosperity* P_i . The door of the restaurant has $K + 1$ states of openness expressed by the integers in the range $[0, K]$. The state of openness can change by one in one unit of time; i.e. it either opens by one, closes by one or remains the same. At the initial moment of time the door is closed (state 0). The i -th gangster enters the restaurant only if the door is opened specially for him, i.e. when the state of openness coincides with his *stoutness* S_i . If at the moment of time when the gangster comes to the restaurant the state of openness is not equal to his stoutness, then the gangster goes away and never returns.

The restaurant works in the interval of time $[0, T]$.

The goal is to gather the gangsters with the maximal total prosperity in the restaurant by opening and closing the door appropriately.

Input

The first line of the input is an integer M , then a blank line followed by M datasets. There is a blank line between datasets.

- The first line of each dataset contains the values N , K , and T , separated by spaces. ($1 \leq N \leq 100$, $1 \leq K \leq 100$, $0 \leq T \leq 30000$)
- The second line of each dataset contains the moments of time when gangsters come to the restaurant T_1, T_2, \dots, T_N , separated by spaces. ($0 \leq T_i \leq T$ for $i = 1, 2, \dots, N$)
- The third line of each dataset contains the values of the prosperity of gangsters P_1, P_2, \dots, P_N , separated by spaces. ($0 \leq P_i \leq 300$ for $i = 1, 2, \dots, N$)
- The fourth line of each dataset contains the values of the stoutness of gangsters S_1, S_2, \dots, S_N , separated by spaces. ($1 \leq S_i \leq K$ for $i = 1, 2, \dots, N$)

All values in the input file are integers.

Output

For each dataset, print to the output a single integer — the maximal sum of prosperity of gangsters in the restaurant. In case when no gangster can enter the restaurant the output should be '0'.

Print a blank line between datasets.

Sample Input

```
2

4 10 20
10 16 8 16
10 11 15 1
10 7 1 8

2 17 1000
5 0
50 33
6 1
```

Sample Output

```
26

0
```