Problem A
AB to C

Time limit: 1 second
Memory limit: 1 GB

Problem Description

You are given a string consisting only of characters A, B, or C. You can do the following operation as many times as you like:

- Choose $1 \leq i < j \leq |S|$ such that $S_i \neq S_j$.
  - Delete S_i and S_j.
  - Insert the complementary character of (S_i, S_j) back into the string at **any** position of your choice.

The complementary character of X is defined as follows:

- If $X = (A, B)$ or (B, A), the complementary character is C.
- If $X = (B, C)$ or (C, B), the complementary character is A.
- If $X = (A, C)$ or (C, A), the complementary character is B.

You are allowed to do as many operations on S as you want. Find the **lexicographically minimal** final string possible.

Input Format

- The first line of input will contain a single integer T, denoting the number of test cases.
- Each test case consists of multiple lines of input.
  - The first line of each test case contains a single integer N - the length of the string.
  - The second line of each test case contains a string S.

Output Format

For each test case, output on a new line the lexicographically minimal string possible after applying operations on S.

Constraints

- $1 \leq T \leq 10^5$
- $1 \leq N \leq 2 \cdot 10^5$
- $S_i \in \{A, B, C\}$
- $|S| = N$
- The sum of N over all test cases does not exceed 10^6.

Samples

Sample Input 1

```
3
2
BC
3
AAA
3
CAA
```

Sample Output 1

```
A
AAA
AB
```

Sample Explanation

Test Case 1 : We use one operation with $i = 1, j = 2$. We delete S_1 and S_2, making the string empty temporarily. The complementary character is A, and we insert that into the string's 1st position. Thus, we end up with A.

Test Case 2 : No operations are possible.

Problem B
Bin Packing

Time limit: 5 seconds
Memory limit: 1 GB

Problem Description

Let's solve a NP Hard problem today.

There are N objects, the i-th of which has size A_i ($1 \leq A_i \leq 12$). You have several bins of size 12. Multiple objects can be fit into the same bin if and only if the sum of their sizes is at most 12.

Find the minimum number of bins needed to fit all N objects.

There is a **special constraint** on the input.

1. For all non-sample files, the number of test cases T and the number of objects N are fixed. ($T = 100, N = 1000$).
2. The object sizes are generated randomly. Formally, each A_i is independently chosen from a uniform distribution on the set of integers in the interval $[1, 12]$.

Input Format

- The first line of input will contain a single integer T, denoting the number of test cases.
- Each test case consists of two lines of input.
  - The first line of each test case contains N - the number of objects.
  - The second line contains N integers - $A_1, A_2, \ldots, A_N$ - the sizes of the objects.

Output Format

For each test case, output on a new line the minimum number of bins needed.

Constraints

For all non-sample tests:

- $T = 100$
- $N = 1000$
- $1 \leq A_i \leq 12$
- Each A_i is a uniformly randomly generated integer in the range $[1, 12]$.

There are exactly 19 non-sample tests.

Samples

Sample Input 1

```
10
5
8 2 2 3 9
3
11 1 1
8
9 8 2 3 4 2 2 12
10
2 2 2 2 2 2 3 3 3 3
6
5 2 8 4 1 11
8
12 12 9 8 2 8 4 4
5
5 5 6 6 6
8
10 12 5 4 8 2 2 11
7
5 1 5 7 9 4 4
6
12 1 1 9 3 5
```

Sample Output 1

```
2
2
4
2
3
5
3
5
4
3
```

Sample Explanation

Test case 1: We can distribute the objects like this:

- Bin 1: Objects 1, 2 and 3. The sum of their sizes is exactly 12.
- Bin 2: Objects 4 and 5. The sum of their sizes is also exactly 12.

Thus, 2 bins suffice.

Problem C
Construct uwu

Time limit: 4 seconds
Memory limit: 1 GB

Problem Description

You are given a positive integer N.

Construct a **minimal** length string S consisting of letters `'u'` and `'w'` only, satisfying the following condition:

- The number of subsequences of S which equal `"uwu"` is exactly N. Note that a subsequence need not be continuous.

It can be proven that at least one valid string exists. You have to find the shortest possible string for each test. The input is generated in such a way that the sum of the lengths of the minimal strings is at most 10^7 over all test cases.

Input Format

- The first line of input contains a single integer T, denoting the number of test cases.
- The first and only line of each test case contains N — the required number of subsequences.

Output Format

For each test case, output a **minimal** length string S with exactly N `"uwu"` subsequences.

Constraints

- $1 \leq T \leq 10^3$
- $1 \leq N \leq 10^{18}$
- The sum of the minimum lengths of the required strings does not exceed 10^7.

Samples

Sample Input 1

```
6
1
2
3
4
5
6
```

Sample Output 1

```
uwu
uwwu
uwwwu
uwwuu
uwwwwwu
uwwwuu
```

Sample Explanation

Test Case 1: `"uwu"` has exactly one subsequence which equals `"uwu"`, which is the whole string itself.

Test Case 2: `"uwwu"` has 2 subsequences which equal `"uwu"`, using 1-based indexing:

- The subsequence formed by indices $(1, 2, 4)$.
- The subsequence formed by indices $(1, 3, 4)$.

Problem D
Counting Distance Arrays

Time limit: 2 seconds
Memory limit: 1 GB

Problem Description

Given a tree with N nodes numbered 1 to N, and a subset of nodes S, we define an array A of size N in the following way:

- $A_i = \max_{x \in S}(\text{dist}(x, i))$.

Here, $\text{dist}(x, y)$ represents the number of edges on the unique shortest path between x and y.

There are $2^N - 1$ non-empty subsets S possible. Find the number of possible distinct arrays A over all $2^N - 1$ choices of S. Since the answer may be large, output it modulo 998244353.

Input Format

- The first line of input contains a single integer T, denoting the number of test cases.
- Each test case consists of multiple lines of input.
  - The first line of each test case contains N — the number of nodes in the tree.
  - The next $N - 1$ lines each contain 2 integers u and v — representing an edge (u, v) in the tree.

Output Format

For each test case, output on a new line the number of distinct arrays A possible over all $2^N - 1$ non-empty subsets S, modulo 998244353.

Constraints

- $1 \leq T \leq 10^4$
- $2 \leq N \leq 2 \cdot 10^5$
- $1 \leq u, v \leq N$
- The set of $N - 1$ input edges form a tree.
- The sum of N does not exceed $2 \cdot 10^5$ over all test cases.

Samples

Sample Input 1

```
3
2
1 2
3
1 2
2 3
7
1 2
2 3
2 4
3 5
1 6
5 7
```

Sample Output 1

```
3
6
23
```

Sample Explanation

Test Case 1: There are 3 subsets S possible. Each of them produces a distinct result, enumerated as follows:

- $S = \{1\}$, the array $A = [0, 1]$
- $S = \{2\}$, the array $A = [1, 0]$
- $S = \{1, 2\}$, the array $A = [1, 1]$

Problem E
Easy Counting Problem

Time limit: 3 seconds
Memory limit: 1 GB

Problem Description

Given an array A, you may perform the following operation zero or more times:

- Choose an index i ($1 \leq i \leq |A|$) such that $A_i = i$ and delete the i-th element from the array A.
- Formally, replace A with $A[1, i-1] + A[i+1, |A|]$ where $A[L, R]$ represents the subarray $[A_L, A_{L+1}, A_{L+2}, ..., A_R]$ and $+$ represents array concatenation.

Let $f(A)$ denote the **minimum possible length** of A after applying operations optimally.

You are given integers N and M, find the sum of $f(A)$ over all M^N arrays of size N and satisfying $1 \leq A_i \leq M$ for all $1 \leq i \leq N$. Since the answer may be large, output it modulo 998244353.

Input Format

- The first line of input will contain a single integer T, denoting the number of test cases.
- The first and only line of input for each test case contains 2 integers N and M.

Output Format

For each test case, output on a new line the sum of $f(A)$ over all arrays of length N and elements bounded by M, modulo 998244353.

Constraints

- $1 \leq T \leq 10$
- $1 \leq N, M < 998244353$
- $|N - M| \leq 2000$

Samples

Sample Input 1

```
9
2 2
2 1
3 1
3 2
3 3
3 5
6 3
23 10
900000000 900000000
```

Sample Output 1

```
3
0
0
9
44
284
2534
243195864
193285282
```

Sample Explanation

Test case 1**:** There are $2^2 = 4$ possible arrays. Their respective values are:

- $f([1, 1]) = 0$. First apply the operation with $i = 1$. Then, A becomes $[1]$, and we can apply the operation with $i = 1$ again, resulting in an empty array.
- $f([1, 2]) = 0$. First apply the operation with $i = 2$, and then with $i = 1$.
- $f([2, 1]) = 2$. No operations are possible.
- $f([2, 2]) = 1$. Apply the operation with $i = 2$.

The sum is 3.

Problem F
Greedy Prices

Time limit: 2 seconds
Memory limit: 1 GB

Problem Description

There are N items for sale at an auction. However, the auction is greedy and has made the prices variable depending on how much it perceives you will be able to pay. If you have already spent some amount of money, then it will increase the cost proportionally, as you are more likely to be willing to buy the item even at a higher price.

Formally, the i-th item has 2 parameters M_i and C_i, and the cost of the item is $M_i \cdot X + C_i$, where X is the **amount of money** you have already spent.
Each item can be bought at most once. You can decide the order of buying items.

You have to answer Q queries of the following form:

- If you had a budget of P_i, what is the maximum number of items you can buy?

Input Format

- The first line of input contains a single integer T, denoting the number of test cases.
- Each test case consists of multiple lines of input.
  - The first line of each test case contains 2 integers — N and Q, the number of items and the number of queries.
  - The second line contains N integers — $M_1, M_2, \ldots, M_N$, the linear coefficients of the items.
  - The third line contains N integers — $C_1, C_2, \ldots, C_N$, the constant coefficients of the items.
  - The next Q lines each contain 1 integer — P_i, representing a budget query.

Output Format

For each test case, output Q integers - the answer to the queries in order.

Constraints

- $1 \leq T \leq 10^4$
- $1 \leq N, Q \leq 2 \cdot 10^5$
- $0 \leq M_i, C_i, P_i \leq 10^9$
- The sum of N and the sum of Q both do not exceed $2 \cdot 10^5$ over all test cases.

Samples

Sample Input 1

```
2
3 4
1 1000 0
3 6 4
2
6
7
19
6 6
0 0 1 1 2 3
0 1 0 2 3 4
0
1
8
15
3
10000
```

Sample Output 1

```
0 1 2 3
2 3 4 5 4 6
```

Sample Explanation

Test Case 1: Here are the answers to the respective queries:

- Budget 2: Impossible to buy any item.
- Budget 6: We can buy any of the 3 items for the prices $3, 6, 4$ respectively. However, we cannot buy more than 1 item.
- Budget 7: First, buy the 1^{st} item for a cost of 3, and then buy the 2^{nd} item for 4.
- Budget 19: Buy all 3 of the items in the following order:
  - $X = 0$, Buy item 2 for a cost of 6.
  - $X = 6$, Buy item 1 for a cost of $1 \cdot 6 + 3 = 9$.
  - $X = 15$, Buy item 3 for a cost of $0 \cdot 15 + 4$.
  - We bought all 3 items spending exactly 19.

Problem G
Lexicographic Raffle

Time limit: 2 seconds
Memory limit: 1 GB

Problem Description

You have a string S of length N consisting of lower-case English characters.
The process `raffle`(L, R) for $1 \leq L < R \leq N$ is defined as follows:

- **Step 1**: Let X be the substring $S_L S_{L+1} \ldots S_{R-1}$ and Y be the substring $S_{L+1} S_{L+2} \ldots S_R$.
- **Step 2**: If Y is **lexicographically smaller**[†] than X, increment L by 1. Otherwise, decrement R by 1.
- **Step 3**: If $L = R$, terminate the process and return L as the result of the process; otherwise, go back to Step 1.

You are given Q queries, each of which contains two integers L and R ($1 \leq L < R \leq N$). For each query, find the final value of L in the process `raffle`(L, R) defined as above.

Input Format

- The first line of input will contain a single integer T, denoting the number of test cases.
- Each test case consists of multiple lines of input:
  - The first line of each test case contains N and Q - the length of the string and the number of queries.
  - The second line of each test case contains S - a string of size N.
  - The next Q lines each contain 2 integers L and R - representing a query.

Output Format

For each test case, for each query, output on a new line the final value of L after the process `raffle`(L, R).

Constraints

- $1 \leq T \leq 10^4$
- $2 \leq N \leq 2 \cdot 10^5$
- $1 \leq Q \leq 2 \cdot 10^5$
- $|S| = N$
- S contains only lower-case English characters.
- $1 \leq L < R \leq N$
- The sum of N and the sum of Q over all test cases both do not exceed $2 \cdot 10^5$.

Samples

Sample Input 1

```
4
6 3
kanpur
1 6
4 6
5 6
10 2
adccbabbab
1 10
2 9
5 3
accaa
1 5
2 4
3 5
5 1
jddda
3 4
```

Sample Output 1

```
2
4
6
1
6
1
4
4
3
```

Sample Explanation

Test Case 1: Here are the explanations for the first query:

- **Query 1**:
  - Initially, $L = 1, R = 6$; $X =$ `kanpu`, $Y =$ `anpur`. Since Y is lexicographically smaller, we increment L to 2. The process does not terminate here since $L \neq R$.
  - Now, $L = 2, R = 6$; $X =$ `anpu`, $Y =$ `npur`. Since Y is **not** lexicographically smaller, we decrement R to 5. The process does not terminate here since $L \neq R$.
  - Now, $L = 2, R = 5$; $X =$ `anp`, $Y =$ `npu`. Since Y is **not** lexicographically smaller, we decrement R to 4. The process does not terminate here since $L \neq R$.
  - Now, $L = 2, R = 4$; $X =$ `an`, $Y =$ `np`. Since Y is **not** lexicographically smaller, we decrement R to 3. The process does not terminate here since $L \neq R$.
  - Now, $L = 2, R = 3$; $X =$ `a`, $Y =$ `n`. Since Y is **not** lexicographically smaller, we decrement R to 2. The process terminates here as $L = R$.

Therefore, the final values are $L = R = 2$. Hence, the answer is 2.

Problem H
Majority Graph

Time limit: 4 seconds
Memory limit: 1 GB

Problem Description

You are given an array A with N elements - $A_1, A_2, \ldots, A_N$.

Construct a graph G on N nodes in the following way:

- Draw an edge (i, j) if and only if the following conditions are satisfied:
  - $1 \leq i < j \leq N$
  - The subarray $A[i, j]$ has a **majority element**.

Find the number of connected components of G.

The subarray $A[i, j]$ refers to the array $[A_i, A_{i+1}, \ldots, A_j]$.
An array B of length M has a **majority element** if and only if there is some element X that occurs **strictly more** than $\frac{M}{2}$ times in B.

Input Format

- The first line of input will contain a single integer T, denoting the number of test cases.
- Each test case consists of two lines of input.
  - The first line of each test case contains a single integer N - the size of the array and graph.
  - The second line contains N integers - $A_1, A_2, \ldots, A_N$.

Output Format

For each test case, output on a new line the number of connected components of the graph G.

Constraints

- $1 \leq T \leq 10^5$
- $2 \leq N \leq 2 \cdot 10^6$
- $1 \leq A_i \leq N$
- The sum of N over all test cases does not exceed $2 \cdot 10^6$.
- Note that the **constraint on N is higher than usual**.

Samples

Sample Input 1

```
4
4
1 2 1 2
5
1 2 3 2 1
2
1 1
3
2 2 1
```

Sample Output 1

```
2
4
1
1
```

Sample Explanation

Test Case 1: There are 2 pairs (i, j) that satisfy the majority condition: $(1, 3)$ and $(2, 4)$. The subarray $A[1, 3]$ has 1 as a majority element, and $A[2, 4]$ has 2. Thus, the graph has 2 connected components.

Test Case 2: There is only one edge $(2, 4)$. Hence, there are 4 connected components.

Problem I
Majority Voters

Time limit: 1 second
Memory limit: 1 GB

Problem Description

There are N voters in a line numbered from 1 to N. They are voting in an election between 2 candidates, Alice and Bob. In order from 1 to N, they will each vote for either Alice or Bob.

The voters' preferences are represented by the string S: $S_i = A$ if the i-th voter was going to vote for Alice, and $S_i = B$ if the i-th voter was going to vote for Bob.

You want to rig the elections in favor of Alice. To do this, you have gained a special power. You can persuade people to change their stance to "majority voter". A majority voter numbered X will vote for the candidate who has already received strictly more votes (from previous voters 1 to $X - 1$). In case both candidates have an equal number of votes, a majority voter will not cast their vote at all.

Alice wins the election if and only if the number of voters for her is strictly greater than the number of voters for Bob.
Let $f(S)$ denote the minimum number of people whom you have to transform to majority voters, for Alice to win the election. If it is impossible for Alice to win no matter what, $f(S) = -1$.

Now, coming to the problem. You are given a string S, containing only characters A and B, of length N, and Q queries of the form:

- Given L and R, find $f(S[L, R])$ where $S[L, R]$ denotes the substring $S_L S_{L+1} \dots S_R$.

Input Format

- The first line of input will contain a single integer T, denoting the number of test cases.
- Each test case consists of multiple lines of input:
  - The first line of each test case contains 2 integers, N and Q - the length of the string and the number of queries.
  - The second line contains a string S of size N.
  - The next Q lines each contain 2 integers L and R - the parameters of each query.

Output Format

For each test case, for each query, output on a new line the value of $f(S[L, R])$.

Constraints

- $1 \leq T \leq 10^4$
- $1 \leq N, Q \leq 2 \cdot 10^5$
- $S_i \in \{A, B\}$
- $|S| = N$

- $1 \leq L \leq R \leq N$
- The sum of N and the sum of Q both do not exceed $2 \cdot 10^5$.

Samples

Sample Input 1

```
2
4 4
AABA
1 4
2 3
3 3
3 4
5 2
BBBBA
1 5
5 5
```

Sample Output 1

```
0
1
-1
1
4
0
```

Sample Explanation

Test Case 1: Here are the explanations for each query.

- **Query 1**: Alice is already winning as she has 3 voters voting for her while Bob has only 1.
- **Query 2**: Alice and Bob are tied with 1 vote each. If you change the Bob voter to a majority voter, Alice ends up winning because the majority voter will also vote for her (since the previous voter had voted for Alice).
- **Query 3**: Best you can do is change the only Bob voter to a majority voter, who will not vote for anyone at all; but that's not sufficient to make Alice win. Hence, the answer is -1.
- **Query 4**: Alice and Bob have one vote each. Changing the one Bob voter to a majority voter will result in Alice's victory, since the majority voter won't vote for anyone while Alice will still have one voter.

Problem J
Max Mod

Time limit: 1.5 seconds
Memory limit: 1 GB

Problem Description

You have an array A of N integers. Initially, all $A_i = 0$. You are also given a **prime** M.

Handle updates and queries of the following format:

1. Given X, Y: add $X + (i-1) \cdot Y$ to A_i for all $1 \le i \le N$.
2. Given L, R: find the maximum value of $(A_i \bmod M)$ across all $L \le i \le R$.

Input Format

- The first line of each test case contains 3 integers - N, M and Q.
- The next Q lines each contain 3 integers in one of the 2 following formats:
  - $T = 1, X, Y$: representing an update with parameters X and Y.
  - $T = 2, L, R$: representing a query with parameters L and R.

Output Format

For each test case, output on a new line the maximum value of $(A_i \bmod M)$ for $L \le i \le R$ for each query.

Constraints

- $1 \le N, Q \le 5 \cdot 10^5$
- $1 \le M \le 10^9$
- M is prime
- $1 \le T \le 2$
- $1 \le X, Y \le 10^9$
- $1 \le L \le R \le N$

Samples

Sample Input 1

```
5 37 6
1 2 7
1 9 13
2 2 4
1 29 3
2 1 3
2 3 5
```

Sample Output 1

```
34
26
35
```

Sample Explanation

Test Case 1: The array changes in the first 3 steps as follows:

- **Update 1**: The array is updated to $[2, 9, 16, 23, 30]$.
- **Update 2**: The array is updated to $[11, 31, 51, 71, 91]$.
- **Query 1**: $A_2 \pmod{37} = 14$, $A_3 \pmod{37} = 34$, $A_4 \pmod{37} = 17$. Thus, the maximum is 34.

Problem K
P to Q

Time limit: 1 second
Memory limit: 1 Gigabyte

Problem Description

You are given 2 permutations P and Q of the integers 1 to N. Your score S is initialized as $\texttt{inversions}(P)^{\dagger}$.

You can do the following operation at most $10 \cdot N$ times:

- Select any integer x satisfying $1 \leq x \leq N$. Delete x from P, and then insert x back into P in any desired location. Replace S with $\max(S, \texttt{inversions}(P))$.

Your goal is to change P into Q while minimizing the final value of S. Also, print the operations performed.
Note that you **do not have to minimize** the number of operations performed.

---

† $\texttt{inversions}(P)$ represents the number of pairs (i, j) satisfying $1 \leq i < j \leq N$ and $P_i > P_j$.

Input Format

- The first line of input will contain a single integer T, denoting the number of test cases.
- Each test case consists of three lines of input:
  - The first line of each test case contains N - the permutation length.
  - The second line contains N integers - $P_1, P_2, \ldots, P_N$ representing the permutation P.
  - The third line contains N integers - $Q_1, Q_2, \ldots, Q_N$ representing the permutation Q.

Output Format

For each test case, the output is as follows:

- Firstly, on a new line, output K $(0 \leq K \leq 10 \cdot N)$ - the number of operations you wish to perform.
- Each of the next K lines must contain 2 space-separated integers:
  - x $(1 \leq x \leq N)$: the number we choose to delete and reinsert in this operation; and
  - pos $(1 \leq pos \leq N)$: the position where we reinsert x. Note there are $N - 1$ elements left after deleting x. $pos = i$ means we will reinsert x right before the i-th element, and $pos = N$ means that we will insert it at the end.

Constraints

- $1 \leq T \leq 100$
- $1 \leq N \leq 500$
- $1 \leq P_i, Q_i \leq N$
- $P_i \neq P_j$ for all $i \neq j$
- $Q_i \neq Q_j$ for all $i \neq j$
- The sum of N over all test cases does not exceed 500.

Samples

Sample Input 1

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

Sample Output 1

```
1
1 2
0
2
1 1
3 1
```

Sample Explanation

Test Case 1: Initially, $S = 0$ because $\texttt{inversions}([1, 2]) = 0$.
We perform one operation, which is to delete 1 and re-insert at the 2nd position.
This changes the permutation to $[2, 1]$ and S gets updated to $\max(0, \texttt{inversions}([2, 1])) = 1$.

We have successfully converted P to Q, and it can be proven that it is impossible to have a lower score.

Test Case 3: We show the steps:

- Initial conditions: $P = [2, 1, 3]$, $S = 1$.
- Delete 1 and reinsert at position 1: $P = [1, 2, 3]$, $S = 1$.
- Delete 3 and reinsert at position 1: $P = [3, 1, 2]$, $S = 2$.

Problem L
Score Sum

Time limit: 1 second
Memory limit: 1 GB

Problem Description

Suppose we have an array A of N elements. We will define a *score* function on the array A in the following way.

Let $d(L, R)$ denote the number of distinct elements in the subarray $[A_L, A_{L+1}, \ldots, A_R]$.
Define $f(L, R) = (R - L + 1) - d(L, R)$, i.e., the length of the subarray from L to R minus the number of distinct elements in it.

Consider the pair (L, R) satisfying $1 \leq L \leq R \leq N$ with the maximum value of $f(L, R)$. If there are multiple such pairs, pick the one with the **minimum** value of $(R - L + 1)$. If there are still multiple such pairs, pick any.

Finally, we define $score(A)$ to be $(R - L + 1)$. That is, the length of the subarray which has the maximum f value and the minimum length among all with maximum f.

Since finding the score was too easy, we added sum over all subarrays as a harder task.

Formally, given an array A of N elements, find the value

$$\sum_{L=1}^{N} \sum_{R=L}^{N} score(A[L, R])$$

where $A([L, R])$ represents the subarray $[A_L, A_{L+1}, \ldots, A_R]$.

Input Format

- The first line of input will contain a single integer T, denoting the number of test cases.
- Each test case consists of two lines of input:
  - The first line of each test case contains N - the size of the array A.
  - The second line of each test case contains N integers - $A_1, A_2, \ldots, A_N$.

Output Format

For each test case, output on a new line the sum of scores of all subarrays of A.

Constraints

- $1 \leq T \leq 10^4$
- $1 \leq N \leq 2 \cdot 10^5$
- $1 \leq A_i \leq N$
- The sum of N over all test cases does not exceed $2 \cdot 10^5$.

Samples

Sample Input 1

```
4
2
1 1
3
1 3 1
5
1 3 1 2 3
7
1 2 3 1 5 3 1
```

Sample Output 1

```
4
8
26
62
```

Sample Explanation

Test Case 1: There are 3 total subarrays - 2 repeated occurrences of [1] and 1 occurrence of $[1, 1]$.

- [1]: $score([1])$ is clearly just 1, because there is only one pair to choose $L = 1, R = 1$.
- $[1, 1]$: $L = 1, R = 2$ is the unique interval with the largest value of f. Hence, $score([1, 1]) = 2$.

The sum is $1 + 1 + 2 = 4$.

Problem M
Ticket Revenue Maximization

Time limit: 4 seconds
Memory limit: 1 GB

Problem Description

There is going to be a tournament with 128 teams numbered $1, 2, \ldots, 128$. For each $i < j$, the team j is stronger than the team i, and in a match between them, the team j always wins. For each i, the team i has P_i supporters. No two teams have any common supporters.

There will be 7 rounds in the tournament. In each round, the teams will be divided into pairs, such that each team belongs to **exactly one** pair. Each pair will play a match, with the loser being eliminated and the winner proceeding to the next round (if any).
Formally,

- In the first round, the 128 teams will be paired into 64 matches. Hence, 64 teams will be eliminated, and the rest will move to the next round.
- In the second round, the 64 winning teams from the first round will be paired into 32 matches.
- In the third round, the 32 winning teams from the second round will be paired into 16 matches.
- and so on, until the final (seventh) round, where the two remaining teams will play a match.

Naturally, the later rounds are more valuable. Hence, the ticket of the $r^{\texttt{th}}$ round costs r. A match between teams i and j in the $r^{\texttt{th}}$ round will therefore create a revenue of $r \cdot (P_i + P_j)$, as each supporter of any of the two teams will buy the tickets for the match, each costing r.

You are in-charge of the pairings in all the rounds, and your objective is to select the pairings in such a way that the total revenue generated through the tournament is maximized. Find this maximum revenue.

Input Format

- The first line of input will contain a single integer T, denoting the number of test cases.
- Each test case contains a single line of input which has 128 integers - $P_1, P_2, \ldots, P_{128}$.

Output Format

For each test case, output on a new line the maximum possible revenue.

Constraints

- $1 \leq T \leq 5$
- $1 \leq P_i \leq 10^9$

Samples

Sample Input 1

```
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 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 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 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 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
128 127 126 125 124 123 122 121 120 119 118 117 116 115 114 113 112 111
110 109 108 107 106 105 104 103 102 101 100 99 98 97 96 95 94 93 92 91
90 89 88 87 86 85 84 83 82 81 80 79 78 77 76 75 74 73 72 71 70 69 68 67
66 65 64 63 62 61 60 59 58 57 56 55 54 53 52 51 50 49 48 47 46 45 44 43
42 41 40 39 38 37 36 35 34 33 32 31 30 29 28 27 26 25 24 23 22 21 20 19
18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
```

Sample Output 1

```
494
30318
```

Note

The sample input 1 in this document has been split into multiple lines to fit them on the page. In reality, all the 128 numbers are present on a single line in the tests.

Sample Explanation

Test case 1: All matches have revenue of $r \cdot 2$ where r is the round number, since all $P_i = 1$.

- Round 1: 64 matches, revenue $64 \cdot 2 = 128$
- Round 2: 32 matches, revenue $32 \cdot 4 = 128$
- Round 3: 16 matches, revenue $16 \cdot 6 = 96$
- Round 4: 8 matches, revenue $8 \cdot 8 = 64$
- Round 5: 4 matches, revenue $4 \cdot 10 = 40$
- Round 6: 2 matches, revenue $2 \cdot 12 = 24$
- Round 7: 1 match, revenue $1 \cdot 14 = 14$

The total revenue is $128 + 128 + 96 + 64 + 40 + 24 + 14 = 494$.

Problem N
Yet Another MST Problem

Time limit: 1 second
Memory limit: 1 GB

Problem Description

You are given a connected undirected graph G with N nodes and M edges. Each edge has weight 1.
Some nodes are marked, and the other nodes are unmarked. You are given a binary string S where $S_i = 1$ if and only if the node i is marked.

Let X be the set of all marked nodes. It is guaranteed that X is non-empty. Construct a complete weighted graph H on the subset X. The weight of the edge between u and v is $\texttt{dist}(u, v)^{\dagger}$, where the distance is measured in the original graph G.

Find the total sum of weights of a minimum spanning tree in H.

---

† $\texttt{dist}(u, v)$ is the length of the shortest path between u and v. The length of a path is measured by the number of edges in the path.

Input Format

- The first line of input will contain a single integer T, denoting the number of test cases.
- Each test case consists of multiple lines of input:
  - The first line of each test case contains N and M - the number of nodes and the number of edges.
  - The second line contains a binary string S of size N.
  - Each of the next M lines contains 2 integers, u and v, representing an edge (u, v) in the graph G.

Output Format

For each test case, output on a new line the sum of weights of the minimum spanning tree of the constructed graph H.

Constraints

- $1 \leq T \leq 10^4$
- $2 \leq N \leq 2 \cdot 10^5$
- $(N - 1) \leq M \leq 2 \cdot 10^5$
- $S_i \in \{0, 1\}$
- $|S| = N$
- There exists at least one i such that $S_i = 1$
- $1 \leq u, v \leq N$
- $u \neq v$
- All the M pairs (u, v) are distinct.
- The given graph G is connected.
- The sum of N and the sum of M both do not exceed $2 \cdot 10^5$.

Samples

Sample Input 1

```
7
3 3
101
1 2
2 3
1 3
6 5
100101
1 2
2 3
3 4
3 5
5 6
4 3
0111
1 2
1 3
1 4
7 7
1110111
1 7
1 4
2 4
3 4
4 5
4 6
4 7
2 1
10
1 2
6 9
100111
2 5
4 3
3 5
5 1
1 6
4 2
4 5
6 3
2 3
6 8
110110
3 5
4 2
2 6
5 1
1 6
6 4
4 3
4 5
```

Sample Output 1

```
1
6
4
9
0
3
3
```

Sample Explanation

Test Case 1: There are 2 marked nodes 1 and 3. $\texttt{dist}(1,3) = 1$ as there is a direct edge $(1,3)$. Hence, the weight of the minimum spanning tree is simply 1.

Test Case 2: There are 3 marked nodes 1, 4, and 6. The weights of each edge are listed below:

- $\texttt{dist}(1,4) = 3$
- $\texttt{dist}(1,6) = 4$
- $\texttt{dist}(4,6) = 3$

The minimum spanning tree contains the 1st and the 3rd edges. Hence, the sum of weights is 6.