The problem:

Consider a money system consisting of n coins. Each coin has a positive integer value. Your task is to calculate the number of distinct *ordered* ways you can produce a money sum x using the available coins.

For example, if the coins are $\{2, 3, 5\}$ and the desired sum is 9, there are 3 ways:

- $2+2+5$
- $3+3+3$
- $2+2+2+3$

Input

The first input line has two integers n and x: the number of coins and the desired sum of money.

The second line has n distinct integers $c_1, c_2, \dots, c_n$: the value of each coin.

Output

Print one integer: the number of ways modulo 10^9+7.

Firstly we define $dp[0], dp[1], dp[2] \dots, dp[x]$ for coin set C.

Now when coin set $C = \phi$ $dp[0]=1$ & $dp[i]=0$ $\forall\ 1 \le i \le x,\ i \in \mathbb{N}$

Now suppose we have the solution $dp[0], dp[1] \dots, dp[x]$ for C.

Now consider coin c has been added to C where $c \notin C$ and let $C' = C \cup \{c\}$

and $dp'[0], dp'[1], \dots dp'[x]$ be the solution for C'.

Now $\underbrace{dp'[m]}_{\text{ways to express } m \text{ using } c'} = \underbrace{dp'[m-c]}_{\text{This means we use atleast one } c} + \underbrace{dp[m]}_{\text{No } c \text{ is used.}}$ for $m \geq c$ } So run a loop on $m = 0, 1, \dots$

So making it more formal

$dp_i[j]$ be the number of distinct orderd ways to produce a sum j using the coins $C[1] \dots, C[i]$.

Now

$dp_i[j] = dp_{i-1}[j] + dp_i[j - C[i]]$ $\forall j \geq C[i]$

and

$dp_i[j] = d_{i-1}[j]$ $\forall j < C[i]$

} See in this for $i=0$ I know $dp_0[0]=1$ & $dp_0[i]=0$ $\forall i \neq 0$.

So I can run a loop on j for each i

```
for (int i=1, i<=n, i++)   // 1-indexing
    for (int j = coins[i], j<=x, j++)   (∵ you do nothing for j < coins i
        dp[j]=(dp[j] + dp[j-c[i]]) % mod.
```

And that's it that's your solution.