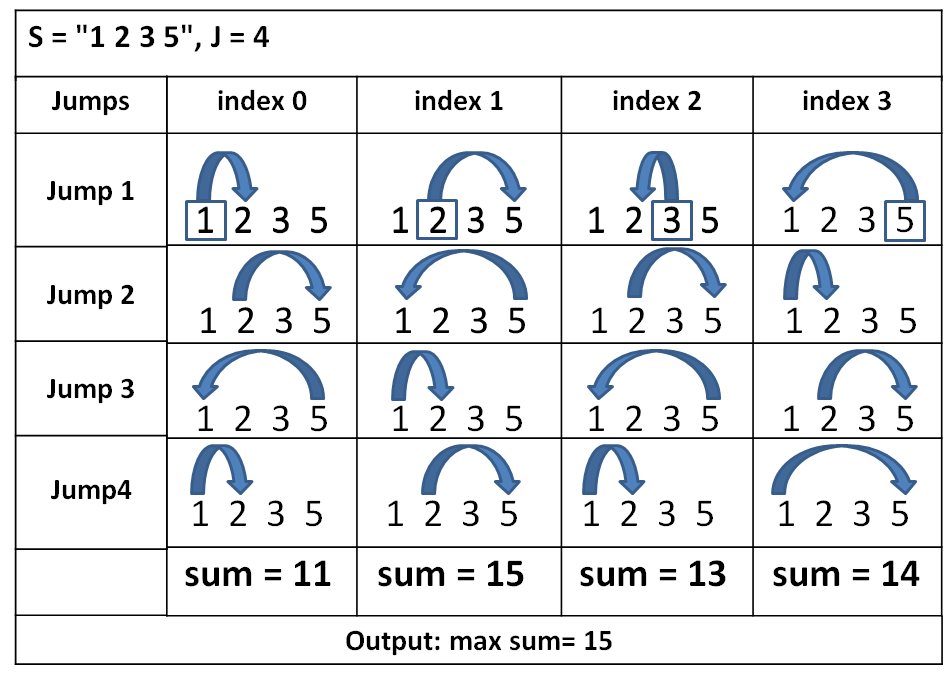
# Problem 2 – Jumping Sums

You are given a string **S** consisting of **N** numbers. Each one of them has indexes ranging from **0 to N-1** and a **value** (integer number). You are also given a number **J** which represents the number of **jumps** to be made. **Each** of the **numbers** jumps **J** times to the right. Your task is to find all **sums** after each of the numbers has made its **J** **jumps.** The **value** of the number and the **current index** show the next index that we have to jump to (e.g. **index = 3, value = 4, nextIndex = 3+4 =7**). Example: if **S =** “**2 1 2”** and **J=2**. The first number “**2**” (**index0**)(**sum1 = sum1 + 2**) should **jump 2** positions to the right, so it jumpsto the last number (**index2**). This index holds value **2** (**sum1 = sum1 + 2**),so it jumps **2** **indexes** to the **right**. In this case, we don’t have enough indexes to the right so we restart from the beginning and the jump will land on **index1** (**sum1 = sum 1 + 1**, **sum1 = 5**). For the second sum we start from the second position (**index1**, **sum2 = sum2 + 1**). The next index is 1 position on the right (**index2**, **sum2 = sum2 + 2**). The next position is 2 positions on the right with restart from the beginning (**index1**, **sum2 = sum2 + 1**). Finally sum2 = 4. We repeat that process for **every** **number** in **S**. Your task is to find the **biggest** **sum** **after** **every** **number** has made **enough** **jumps**. Look at the **picture** on the **right** to understand your task better.

### Input

The input data should be read from the console. It consists of 2 lines:

* On the **first line** you will be given string **S** holding integer numbers separated by a **single space**.
* On the **second line** you will be given **integer J** representing the number of jumps each number should make.

The input data will always be valid and in the format described. There is no need to check it explicitly.

### Output

The output should be printed on the console. It should consist of exactly **1** line:

* On the **first line** print the **maximal** **sum** in the format “**max sum = {0}**”.

### Constraints

* The string **S** will hold maximum of 100 numbers in the range [0…1000]
* The **J** input value will be an integer in the range [1…10000]
* Allowed working time: 0.2 seconds. Allowed memory: 16 MB.

### Examples

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |
| 1 2 3 5  4 | max sum = 15 |  | 165 782 15 77 15 20  102 | max sum = 23229 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |
| 2 7 4 10 6 9  14 | max sum = 90 |  | 1 1 1 1 1 1 1  50 | max sum = 51 |