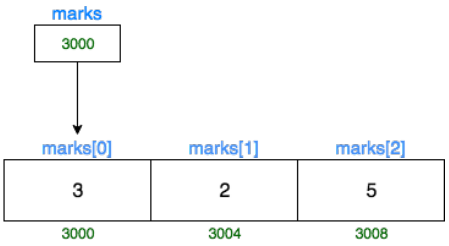
Students Marks Sum

You are given an array of integers, $marks$, denoting the marks scored by students in a class. The alternating elements $marks_0$, $marks_2$, $marks_4$ and so on are the marks of boys whereas $marks_1$, $marks_3$, $marks_5$ and so on are the marks of girls. The array name, $marks$, works as a pointer which stores the base address of that array. In other words, $marks$ contains the address where $marks_0$ is stored in the memory.

For example, let $marks$ = [3, 2, 5] and $marks$ = 3000. Then, 3000 is the memory address of $marks_0$.



Complete the function, `marks_summation(int* marks, char gender, int number_of_students)` which returns the total sum of:

- marks of boys if $gender = b$
- marks of girls if $gender = g$

The locked code stub reads the elements of $marks$ along with $gender$. Then, it calls the function `marks_summation(marks, gender, number_of_students)` to get the sum of alternate elements as explained above and then it prints the sum.

Input Format

- The first line contains $number\_of\_students$, denoting the number of students in the class, hence the number of elements in $marks$.
- Each of the $number\_of\_students$ subsequent lines contains $marks_i$.
- The next line contains $gender$.

Constraints

- $1 \leq number\_of\_students \leq 10^3$
- $1 \leq marks_i \leq 10^3$ (where $0 \leq i < number\_of\_students$)
- $gender = g$ or b

Output Format

The output should contain the sum of all the aternate elements in $marks$ as explained above.

Sample Input 0

```
3
3
2
5
b
```

Sample Output 0

```
8
```

Explanation 0

$marks$ = [3, 2, 5] and $gender$ = b.

So, $marks_0 + marks_2 = 3 + 5 = 8$.

Sample Input 1

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

Sample Output 1

```
6
```

Explanation 1

$marks$ = [1, 2, 3, 4, 5] and $gender$ = g

So, $sum = marks_1 + marks_3 = 2 + 5 = 8$.

Sample Input 2

```
1
5
g
```

Sample Output 2

```
0
```

Explanation 2

$marks$ = [5] and $gender$ = g

Here, $marks_1$ does not exist. So, $sum = 0$.