

I - Implementation Addict

Time limit : 2sec / Memory limit : 256MB

Problem

Wolf Sothe loves implementing online judge problems, he wants to solve as many problems as possible. However, if he keeps solving problems every day without a rest, day by day the number of problems that can be solved in a day will reduce.

So, sometimes Wolf Sothe takes a day off. On that day, Wolf Sothe will not solve any problem, then for the next day and later Wolf Sothe will be able to solve problems.

The number of problems that Wolf Sothe can solve in a day is as follows:

- If that day is a working day, we define the number of days that Wolf Sothe has kept solving problems as k (not including that day). Then Wolf Sothe can solve $\max(0, A - k \times B)$ problems during that day.
- If that day is a rest day, during that day no problem will be solved.

Wolf Sothe wants to solve problems for N days. Let the N days be as a sequence from the 1st day to the N_{th} day. Suppose that Wolf Sothe doesn't solve problems before the 1st day.

In addition, we known in advance that there are some decided rest days.

A list of all the decided rest days will be given. For any other day, you can mark it as a rest day or a working day. Please find the maximum value of the number of problems that can be solved during N days.

Input

Inputs will be given by standard input in following format

```
N A B M
t1
t2
:
tM
```

- For the first line, integer $N(1 \leq N \leq 1,000,000,000)$, $A(1 \leq A \leq 1,000,000,000)$, $B(1 \leq B \leq 1,000,000,000)$, $M(0 \leq M \leq 100,000)$ will be given divided by spaces. M represents the number of the decided rest days.
- From the second line there are M additional lines to give the information about decided rest days. For the i_{th} line, an integer $t_i(1 \leq t_i \leq N)$ that represents the t_i th decided rest day will be given. Please note that if $i < j$, then $t_i < t_j$

Output

Please output the maximum value of the number of problems that can be solved during N days in one line.

Print a newline at the end of output.

Input Example 1

```
5 6 2 0
```

[Copy](#)

Output Example 1

```
20
```

[Copy](#)

Suppose that Wolf Sothe rest on the 3rd day and solve problems on the remaining 4 days.

- For the 1st day, Wolf Sothe has kept solving problems for 0 day. Thus, $\max(0, 6 - 0 \times 2) = 6$ problems.
- For the 2nd day, Wolf Sothe has kept solving problems for 1 day. Thus, $\max(0, 6 - 1 \times 2) = 4$ problems.
- For the 3rd day, Wolf Sothe rests. Thus, 0 problem.
- For the 4th day, Wolf Sothe has kept solving problems for 0 day. Thus, $\max(0, 6 - 0 \times 2) = 6$ problems.
- For the 5th day, Wolf Sothe has kept solving problems for 1 day. Thus, $\max(0, 6 - 1 \times 2) = 4$ problems.

In conclusion, $6 + 4 + 0 + 6 + 4 = 20$ will be solved in total.

Input Example 2

```
6 4 3 1
3
```

[Copy](#)

Output Example 2

```
13
```

[Copy](#)

Input Example 3

```
12 10 3 3
2
```

[Copy](#)

7
10

Output Example 3

71

Copy