Jogging in NUS

Objective

Students are expected to solve the problems using basic looping techniques.

Problem Description

John likes jogging inside the NUS campus. Every time John starts jogging from PGP and he must be back to PGP within **M** seconds (1 <= **M** <= 1,000,000). However, the road in NUS is not always flat, sometimes uphill or downhill. The road can be divided into **T** units (1 <= **T** <= 10,000) in length and consists of equal-length portions that are uphill, flat, or downhill.

John takes **U** seconds (1 <= **U** <= 1000) to run one unit of uphill road, **F** (1 <= **F** <= 1000) seconds for a unit of flat road, and **D** (1 <= **D** <= 1000) seconds for a unit of downhill road. Note that, when returning to PGP, uphill units become downhill units and downhill units become uphill units.

Given the road description and time limit (**M** seconds), help John to figure out the farthest distance (# of units) he can run from PGP and still can be back to PGP within **M** seconds.

Input

Line 1: **M**, **T**, **U**, **F**, and **D** separated by space.

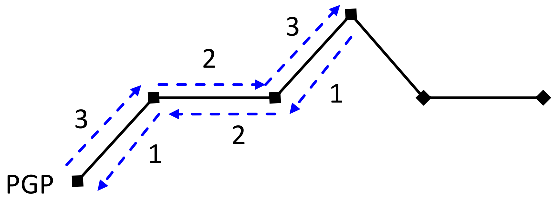
Next **T** lines: Road description. Line i + 1 describes the road unit i using a single character that is u, f, or d, indicating respectively uphill, flat, or downhill.

Output

A single integer that is the farthest distance (# of units) that John can run from PGP and make it back in time.

Sample Input

```
13 5 3 2 1
u
f
u
d
f
```



Sample Output

```
3
```

Note

The main Java class must be called **Jogging**, and be in the source file **Jogging**.java.