

---

## Problem

---

You will be given two DNA sequences for two viruses. The DNA sequences consist of various length of A, T, C and G nucleobases. The viruses are prone to mutations and genetic modifications. You will be given different types of operations on two viruses to generate different types of viruses.

---

## Specifications

---

1. The gene sequence consists of at least 16, at most 128 bases.
  2. The input contains multiple lines of information.
  3. The first 2 lines contains the base sequences for the viruses.
  4. Next 4 lines is the operations defined on these sequences.
  5. Each operation start with the operation code and followed by its arguments.
  6. You are expected to print out the sequences after modifications.
  7. Operations may modify the viruses or make a new virus, so you need three one-dimensional arrays for the viruses.
  8. Initialize all viruses to character value 0, not the character 0.
  9. All given positions start from 0 for both input for output.
  10. The output contains 3 lines ending with a newline character, one for each virus, modified or generated. Make sure to obey this one as a single character difference may result in a wrong answer even if it is invisible.
- The third line may be an empty line if no new virus is made.
11. You will read the input from standard input and print out to the standard output. So, make use of scanf and printf functions.

---

## Operations

---

### 1. Mutate

This basic mutation changes a base into another one.

The input is specified as follows:

Input format: M 1 3 A

1. M: the operation code for mutation
2. 1: the virus -> this is either one of 1, 2 or 3
3. 3: the base position -> from 0 up to the sequence length-1
4. A: the new base for this position -> either one of A, T, C or G

Therefore, given this input you need to change the base in position 3 (fourth from the start) of the first virus into A no matter what the previous base is.

### 2. Flip

Some viruses are not that lucky as the second mutation is not as free as the first one. This time mutations occur between only purine or only pyrimidine bases, therefore A's turn into G's, T's into C's and vice versa.

Input format: F 2 0

1. F: the operation code for flip
2. 2: the virus -> this is either one of 1, 2 or 3
3. 0: the base position -> from 0 up to the sequence length-1

Given this input, you need to change the base in position 0 (the first one) of the virus 2. If it is an A make G, if it is a G make it A and so on.

3. Join

Crazy scientists use this operation to make a new virus. The first part of the new virus is taken from the first virus and the remaining part is taken from the second.

Input format: J 13

1. J: the operation code for flip
2. 13: the split position for viruses

Given this input, you need to combine the first 13 bases of the first virus and the bases after position 13 of the second virus to make the third virus.

---

### Input & Output Examples

---

Remember to obey the specifications for input and output. You may use files with < and > operators to redirect them to standard input and output.

1. Compile your code as below:

```
gcc lab_1.c -o lab_1 -Wall -pedantic-errors
```

2. Run it as follows if you put the input to the 'input\_1.txt'. The output (things that you printf) will be in 'output\_1.txt'.

```
../lab_1 < input_1.txt > output_1.txt
```

3. You can use 'diff' command to compare outputs.

---

#### -----Example 1-----

##### -----Input-----

```
ATCGACGTACGGTGTAACAG
```

```
GCCGTAGTAGGATCCATCGA
```

```
M 1 3 A
```

```
M 2 3 A
```

```
M 1 4 A
```

```
M 2 4 A
```

##### -----Output-----

```
ATCAACGTACGGTGTAACAG
```

```
GCCAAAGTAGGATCCATCGA
```

---

#### -----Example 2-----

##### -----Input-----

```
ATCGACGTACGGTGTAACAG
```

```
GCCGTAGTAGGATCCATCGA
```

```
F 2 17
```

F 2 9  
F 1 0  
F 1 1

-----Output-----  
GCCGACGTACGGTGTAACAG  
GCCGTAGTAAGATCCATTGA

-----Example 3-----  
-----Input-----  
ATCGACGTACGGTGTAACAG  
GCCGTAGTAGGGATCCATCGA  
M 1 3 A  
F 2 11  
J 7  
J 10

-----Output-----  
ATCAACGTACGGTGTAACAG  
GCCGTAGTAGGGTCCATCGA  
ATCAACGTACGGTCCATCGA

-----Example 4-----  
-----Input-----  
ATCGACGTACGGTGTAACAG  
GCCGTAGTAGGGATCCATCGA  
M 1 3 A  
F 2 9  
M 2 15 C  
J 7

-----Output-----  
ATCAACGTACGGTGTAACAG  
GCCGTAGTAAGATCCCTCGA  
ATCAACGTAAGATCCCTCGA

-----End-----