Acadox Problem

Acadox vision is to provide innovative and modern Learning Management technologies that empowers the faculty and students to engage and collaborate in a simple and efficient manner.

One of the professor, who loves Acadox so much, posted on his page as a teaser problem for his students to prepare for their programming exam. The problem posted was as follows:

Develop program that emulates a simple hexadecimal calculator that uses postfix notation. Perform the operations of addition, subtraction, logical and, logical or, logical not, and logical exclusive or.

```
00000000  3C 21 44 4F 43 54 59 50  45 20 48 54 4D 4C 20 50  <!DOCTYPE HTML P
00000010  55 42 4C 49 43 20 22 2D  2F 2F 57 33 43 2F 2F 44  UBLIC "-//W3C//D
00000020  54 44 20 48 54 4D 4C 20  34 2E 30 31 20 54 72 61  TD HTML 4.01 Tra
00000030  6E 73 69 74 69 6F 6E 61  6C 2F 2F 45 4E 22 20 22  nsitional//EN" "
00000040  68 74 74 70 3A 2F 2F 77  77 77 2E 77 33 2E 6F 72  http://www.w3.or
00000050  67 2F 54 52 2F 68 74 6D  6C 34 2F 6C 6F 6F 73 65  g/TR/html4/loose
00000060  2E 64 74 64 22 3E 0D 0A  3C 68 74 6D 6C 3E 0D 0A  .dtd">..<html>..
```

Description

Since Acadox is social environment for learning. The professor posted the following description on his page: The programmer's calculator accepts a string of hexadecimal digits and operators in reverse Polish (postfix) notation then produces the result. Input digits represent 16 bit unsigned binary bit strings.

Input

The program must accept a sequence of operators and hexadecimal digits

following the postfix form, as follows.

Digits: Leading zeros are optional, alphas are not case sensitive: {[0-9 | A-F | a-f]}1-4

Operators :

+	Plus sign	Addition
-	Dash	Subtraction
&	Ampersand	Logical And
|	Vertical bar	Logical Or
~	Tilde	Logical Not
X	Upper case X	Logical Exclusive Or

Each input item is delimited by a white space. An input stream is terminated with a new-line. No more than 20 items are accepted.

Output

The program must display the result of evaluating the entire postfix expression as single hexadecimal string, with leading zeros and upper case letters. If any input is invalid, the string “ERROR” is displayed.

All operations are bitwise – there is no representation of negative quantities. An overflow: x + y > FFFF results in FFFF. An underflow, x – y < 0000, results in 0000.

Sample Input 1:

```
1 1 +
```

Sample Output 1:

```
0002
```

Sample Input 2:

```
F 1 –
```

Sample Output 2:

```
000E
```

Sample Input 3:

```
F - 1
```

Sample Output 3:

```
ERROR
```

Problem Author: IEEE

Suggest Edits

Emacs | Normal | Vim

Select Language: C++

save code

```cpp
#include <cmath>
#include <cstdio>
#include <vector>
#include <iostream>
#include <algorithm>
using namespace std;


int main() {
    /* Enter your code here. Read input from STDIN. Print output to STDOUT */
    return 0;
}

```

Line: 1 Col: 1 Count: 227

Use a custom test case

Upload Code as File

Compile & Test | Submit Code