Write a C program to calculate the result of n expressions written in the form of ***(string1)op(string2)*** where ***op*** represents the operators (+ , - , \*, /, %) and ***string1*** and ***string2*** represent two strings. First you have to check whether the two strings are valid integer constants or not.

If so, apply the operation *op* on both the strings and display the result in a format as per the following rules:

1. (string1) +(string2) - output in base 3 format

2. (string1) -(string2) - output in base 3 format

3. (string1) \*(string2) - output in base 9 format

4. (string1)/(string2) - output (only the quotient) in base 5 format

5. (string1) %(string2) - output in base 7 format

If not, print the following results

- if string1 is valid: print (decimal equivalent of string1)op(error)

- if string2 is valid: print (error)op(decimal equivalent of string2)

- if both strings are invalid: print (error)op(error)

You should define the following functions in your program for doing the calculations

1. **int check (char input [ ])** - checks whether the input strings represent integer constants or not.
2. **int type (char input [ ])** - checks the type of input string (decimal, octal or hexadecimal).
3. **void convert\_to\_base3(char input1[ ], char input2[ ])** - convert input1[] to base 3 format and store the result in input2[].
4. **void convert\_to\_base5(char input1[], char input2[ ])** - convert input1[] to base 5 format and store the result in input2[].
5. **void convert\_to\_base7(char input1[], char input2[ ])** - convert input1[] to base 7 format and store the result in input2[].
6. **void convert\_to\_base9(char input1[], char input2[ ])** - convert input1[] to base 9 format and store the result in input2[].
7. **void add (char input1[], char input2[ ])** - adds the input strings, interpreting them as numeric constants, and stores the result in input1[].
8. **void subtract (char input1[], char input2[ ])** - subtracts input2 from input1, interpreting them as numeric constants, and stores the result in input1[].
9. **void multiply (char input1[], char input2[ ])** - multiplies the input strings, interpreting them as numeric constants, and stores the result in input1[].
10. **void divide (char input1[], char input2[ ])** - divide input1 by input2, interpreting them as numeric constants, and store the **quotient** in input1[].
11. **void remainder (char input1[], char input2[ ])** - find the remainder when input1 is divided by input2, interpreting them as numeric constants, and store the result in input1[].

**Input Format**

The first line represents the number of input expressions n. The following n lines contain one expression each in the format (string1) op(string2). where the maximum length of an expression is 50 characters.

**Constraints**

* (0 < n < 100)
* The operands ***string1*** and ***string2*** can be of different formats (decimal representation or octal representation or hexadecimal representation).
* Not allowed to use any standard library files other than stdio.h

**Output Format**

n lines containing the corresponding output for the input expressions.

**Sample Input 0**

3

(100)+(25)

(100)/(25)

(100)%(25)

**Sample Output 0**

11122

4

0

**Sample Input 1**

4

(0xA)+(12)

(12)+(0xA)

(077)+(100L)

(100KG)+(0XFUL)

**Sample Output 1**

211

211

20001

(error)+(15)

**Sample Input 2**

1

(NIT)/(Calicut)

**Sample Output 2**

(error)/(error)