Chapter VIII

Exercise 05 : do-op

	Exercise 05
do-op	
Turn-in directory : $ex05/$	
Files to turn in : `Your program files`	
Allowed functions : `write`	

- Create a program called `do-op`.
- The progam will be executed with three arguments: `do-op value1 operateur value2`
- Example :

```
$>./do-op 42 "+" 21
63
$>
```

- You should use an array of pointers to function to take care of the `operator`.
- In case of an invalid operator your program should print `0`.
- If the number of arguments is invalid, `do-op` doesn't display anything.
- Your program should accept and print the result for the following operators: '+' '-' '/' '*' and '%'
- In case of a division by 0, it should print:

```
Stop : division by zero
```

- In case of a modulo by 0, it should print:

```
Stop : modulo by zero
```

- Here's an example of tests the Moulinette will run :

```
$> make clean
$> make
$> ./do-op
$> ./do-op 1 + 1
2
$> ./do-op 42amis - --+-20toto12
62
$> ./do-op 1 p 1
0
$> ./do-op 1 + toto3
1
$>
$> ./do-op toto3 + 4
4
$> ./do-op foo plus bar
0
$> ./do-op 25 / 0
Stop : division by zero
$> ./do-op 25 % 0
Stop : modulo by zero
$>
```