

Challenge 2: Make a Calculator

In this exercise, you are required to implement a calculator by using match statements.

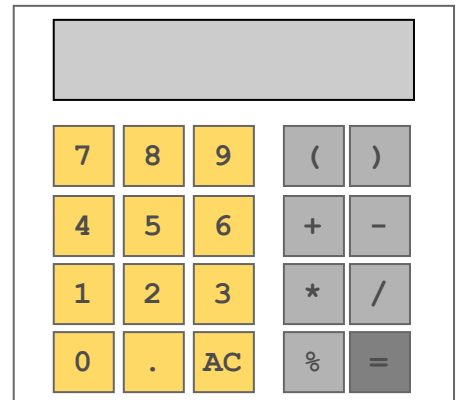
We'll cover the following ^

- Problem Statement
- Sample Input
- Sample Output
- Coding Exercise

Problem Statement

Write a code which will take:

- Two variables named `a` and `b`
- a character type variable called `operator` which will take operators (`+`, `-`, `/`, `*`, `%`) will be passed as input to our `match` statement
- Use `match` statements to compute: addition of `a` and `b`, subtraction of `b` from `a`, multiplication of `a` and `b`, division of `a` by `b`, modulus of `a` by `b`



Sample Input

Suppose three inputs are given:

```
a = 3, b = 2, operator = '+'  
a = 3 , b = 2, operator = '('  
a = 3 , b = 0, operator = '/'
```

Sample Output

The output of the program corresponding to input:

```
5
invalid operator
Division by 0 is undefined
```

Coding Exercise

Write your code below. It is recommended that you try solving the exercise yourself before viewing the solution.

Note: There is a `test` function given in the code for testing purposes. Do not modify it.

Important Note: In the code below you only have to write the `match` construct. Assume that `_a` is the first number, `_operator` is the operator and `_b` is the second number.

For any invalid operator passed as input the program should print invalid

Good luck! 🍀

```
fn test(a: i32, operator: char, b: i32) {
    // Write code here
}
```



Let's move on to the detailed review of the solution.