

## Objective

In this challenge, you will learn simple usage of functions in C. Functions are a bunch of statements grouped together. A function is provided with zero or more arguments, and it executes the statements on it. Based on the return type, it either returns nothing (void) or something.

A sample syntax for a function is

```
return_type function_name(arg_type_1 arg_1, arg_type_2 arg_2, ...) {
    ...
    ...
    ...
    [if return_type is non void]
        return something of type `return_type`;
}
```

For example, a function to read four variables and return the sum of them can be written as

```
int sum_of_four(int a, int b, int c, int d) {
    int sum = 0;
    sum += a;
    sum += b;
    sum += c;
    sum += d;
    return sum;
}
```

`+=` : Add and assignment operator. It adds the right operand to the left operand and assigns the result to the left operand.

`a += b` is equivalent to `a = a + b`;

## Task

Write a function `int max_of_four(int a, int b, int c, int d)` which reads four arguments and returns the greatest of them.

## Note

There is not built in `max` function in C. Code that will be reused is often put in a separate function, e.g. `int max(x, y)` that returns the greater of the two values.

## Input Format

Input will contain four integers - *a, b, c, d* , one on each line.

## Output Format

Print the greatest of the four integers.

Note: I/O will be automatically handled.

## Sample Input

```
3
4
6
5
```

## Sample Output

```
6
```