# 102-A06 Computing a to the power b (POW)

## Work to do

- You will add the following prototype to **tools.h**
  int pow ( int a , int b );
- You will implement the following function in **tools.c**
- You will invoke this function when menu option 2 is chosen from the **main.c** *main* function.
- You will invoke this function from **tests.c** several times with different parameters and test the return value to make sure it's working completely. Consider this as an implementation of a test-harness that will run automatically.

You need to write a function named *pow* which takes two positive integers *a* and *b* and returns *a* to the power *b*. You will have to compute this value by only using multiplications; *a*^*b* means that *a* is multiplied by itself b times.

If one or both of the parameters are non valid (e.g. negative), your function will simply return -1. The code in the *main* functions in **main.c** and **tests.c** will always check if the call went ok by comparing the return value to -1, it it's equal, the parameters were invalid otherwise the function did its job. Make sure you include such scenario in your test harness.

## Example(s)

Here are some examples of return values when calling pow;

| | | |
|---|---|---|
| pow ( 2 , 9 ) | → | 512 |
| pow ( 3 , 3 ) | → | 27 |
| pow ( 30 , 0 ) | → | 1 |
| pow ( -1 , 3 ) | → | -1 |

## Hints

- n/a

## Testing

| **Input** | | **Output** | |
|---|---|---|---|
| a | b | **Expected** | **Observed** |
| | | | |