

SADS 2019 Problem Sheet #2

Problem 2.1: *correctness of exponentiation algorithm* (1+2+2+2+1+1+1 = 10 points)

Prove step-by-step the partial correctness and the total correctness of the function `exp()` using Hoare Logic.

```
#include <stdlib.h>
#include <stdio.h>

static int exp(int x, int n)
{
    int K = n;
    int P = x;
    int Y = 1;

    while (K > 0) {
        if (K % 2 == 0) {
            P = P * P; K = K / 2;
        } else {
            Y = Y * P; K = K - 1;
        }
    }

    return Y;
}

int main(int argc, char *argv[])
{
    if (argc != 3) {
        return EXIT_FAILURE;
    }
    printf("%d\n", exp(atoi(argv[1]), atoi(argv[2])));
    return EXIT_SUCCESS;
}
```

Our claim is that the function `exp(x, n)` calculates x^n for integers x and n .

- Translate the C function into Hoare language constructs and define the precondition and the postcondition of the function `exp()`.
- Add annotations for partial correctness.
- Derive verification conditions for partial correctness.
- Prove the partial correctness verification conditions.
- Add additional annotations for total correctness.
- Derive or update verification conditions for total correctness.
- Prove the total correctness verification conditions.