

Recursion Exercise 1

Recursion Exercise 1

Rewrite the following recursive function in iterative form.

```
unsigned int f (const unsigned int n)
{
    if (n <= 2) return 1;
    return f(n-1) + 2 * f(n-3);
}
```

Recursion Exercise 1

Solution below uses just 4 variables.

Other solutions are of course also possible.

```
unsigned int f_it (const unsigned int n)
{
    if (n <= 2) return 1;
    unsigned int a = 1; // f(0)
    unsigned int b = 1; // f(1)
    unsigned int c = 1; // f(2)
    for (unsigned int i = 3; i < n; ++i) {
        const unsigned int a_prev = a; // f(i-3)
        a = b; // f(i-2)
        b = c; // f(i-1)
        c = b + 2 * a_prev; // f(i)
    }
    return c + 2 * a; // f(n-1) + 2 * f(n-3)
}
```

Recursion Exercise 2

Recursion Exercise 2

Rewrite the following recursive function in iterative form.

```
unsigned int f (const unsigned int n)
{
    if (n == 0) return 1;
    return f(n-1) + 2 * f(n/2);
}
```

Recursion Exercise 2

Solution below stores intermediate results in a **vector**. Other solutions are of course also possible.

```
unsigned int f_it (const unsigned int n)
{
    if (n == 0) return 1; // special case

    std::vector<unsigned int> f_values(n+1, 0);
    f_values[0] = 1;
    for (unsigned int i=1; i<=n; ++i)
        f_values[i] = f_values[i-1] + 2*f_values[i/2];

    return f_values[n];
}
```

Recursion Exercise 3

Recursion Exercise 3

Write a function `rev_out` (see template below) which outputs the contents of an `istream` in reverse order using recursion.

```
#include <iostream>
#include <sstream>

// POST: output the content of is in reverse order to
//       std::cout, and removed it from is.
void rev_out (std::istream& is)
{
    // your code
}

int main () {
    std::stringstream input ("abcdefghijklmno");
    rev_out (input);
    return 0;
}
```


Recursion Exercise 3

Other solutions are of course also possible.

```
#include <iostream>
#include <sstream>

// POST: output the content of is in reverse order to
//       std::cout, and removed it from is.
void rev_out (std::istream& is)
{
    char val;
    if (is >> val) {
        rev_out(is);
        std::cout << val;
    }
}

int main () {
    std::stringstream input ("abcdefghijklmno");
    rev_out (input);
    return 0;
}
```