

# Exercise 1

# Exercise 1

Find **PRE- and POST-conditions** for this function.

## 1. Function:

```
double f (double i,  
          double j,  
          double k)  
{  
    if (i > j) {  
        if (i > k) return i;  
        else return k;  
    } else {  
        if (j > k) return j;  
        else return k;  
    }  
}
```

# Exercise 1

## PRE-Condition:

(not needed)

## POST-Condition:

```
// POST: return value is  
//       the maximum of  
//       i, j and k
```

## 1. Function:

```
double f (double i,  
          double j,  
          double k)  
{  
    if (i > j) {  
        if (i > k) return i;  
        else return k;  
    } else {  
        if (j > k) return j;  
        else return k;  
    }  
}
```

# Exercise 1

Find **PRE- and POST-conditions** for this function.

## 2. Function:

```
double g (int i, int j)
{
    double r = 0.0;
    for (int k = i; k <= j; ++k) {
        r += 1.0 / k;
    }
    return r;
}
```

# Exercise 1

## 2. Function:

```
double g (int i, int j)
{
    double r = 0.0;
    for (int k = i; k <= j; ++k) {
        r += 1.0 / k;
    }
    return r;
}
```

```
PRE-Condition: // PRE: 0 not contained in {i, ..., j} and i <= j
                // and j < INT_MAX
POST-Condition: // POST: return value is the sum
                //          1/i + 1/(i+1) + ... + 1/j
```

# Exercise 2

# Exercise 2

- What is the **output** of this program?
- You can neglect possible over- or underflows for this exercise.

```
#include <iostream>

int f (int i) {
    return i * i;
}

int g (int i) {
    return i * f(i) * f(f(i));
}

void h (int i) {
    std::cout << g(i) << "\n";
}

int main () {
    int i;
    std::cin >> i;
    h(i);
    return 0;
}
```

# Exercise 2

`i * f(i) * f(f(i))`

```
#include <iostream>

int f (int i) {
    return i * i;
}

int g (int i) {
    return i * f(i) * f(f(i));
}

void h (int i) {
    std::cout << g(i) << "\n";
}

int main () {
    int i;
    std::cin >> i;
    h(i);
    return 0;
}
```



# Exercise 2

$i * f(i) * f(f(i))$



$f(i)$

```
#include <iostream>

int f (int i) {
    return i * i;
}

int g (int i) {
    return i * f(i) * f(f(i));
}

void h (int i) {
    std::cout << g(i) << "\n";
}

int main () {
    int i;
    std::cin >> i;
    h(i);
    return 0;
}
```

# Exercise 2

`i * f(i) * f(f(i))`



`i*i`

```
#include <iostream>

int f (int i) {
    return i * i;
}

int g (int i) {
    return i * f(i) * f(f(i));
}


void h (int i) {
    std::cout << g(i) << "\n";
}

int main () {
    int i;
    std::cin >> i;
    h(i);
    return 0;
}
```

# Exercise 2

`i * (i*i) * f(f(i))`

`i*i`



```
#include <iostream>

int f (int i) {
    return i * i;
}

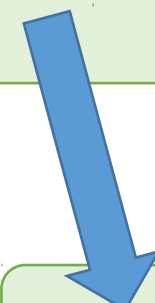
int g (int i) {
    return i * f(i) * f(f(i));
}

void h (int i) {
    std::cout << g(i) << "\n";
}

int main () {
    int i;
    std::cin >> i;
    h(i);
    return 0;
}
```

# Exercise 2

`i * (i*i) * f(f(i))`



`f(f(i))`

```
#include <iostream>

int f (int i) {
    return i * i;
}

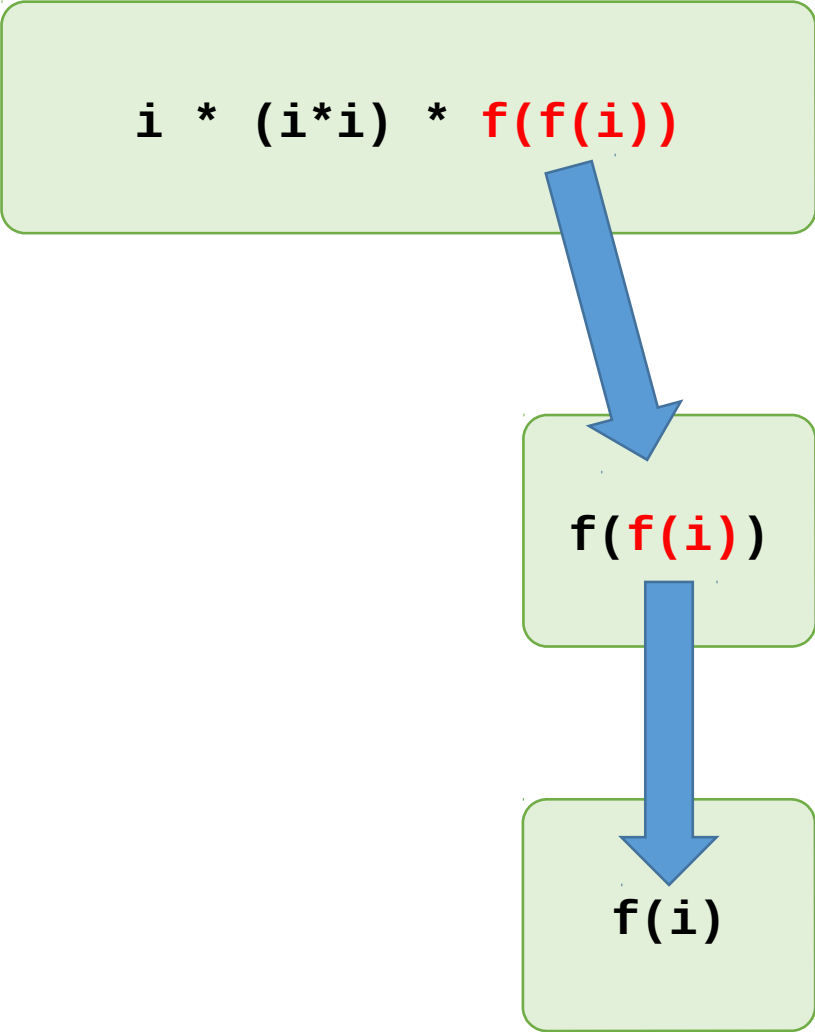
int g (int i) {
    return i * f(i) * f(f(i));
}

void h (int i) {
    std::cout << g(i) << "\n";
}

int main () {
    int i;
    std::cin >> i;
    h(i);
    return 0;
}
```

# Exercise 2

`i * (i*i) * f(f(i))`



`f(f(i))`

`f(i)`

```
#include <iostream>

int f (int i) {
    return i * i;
}

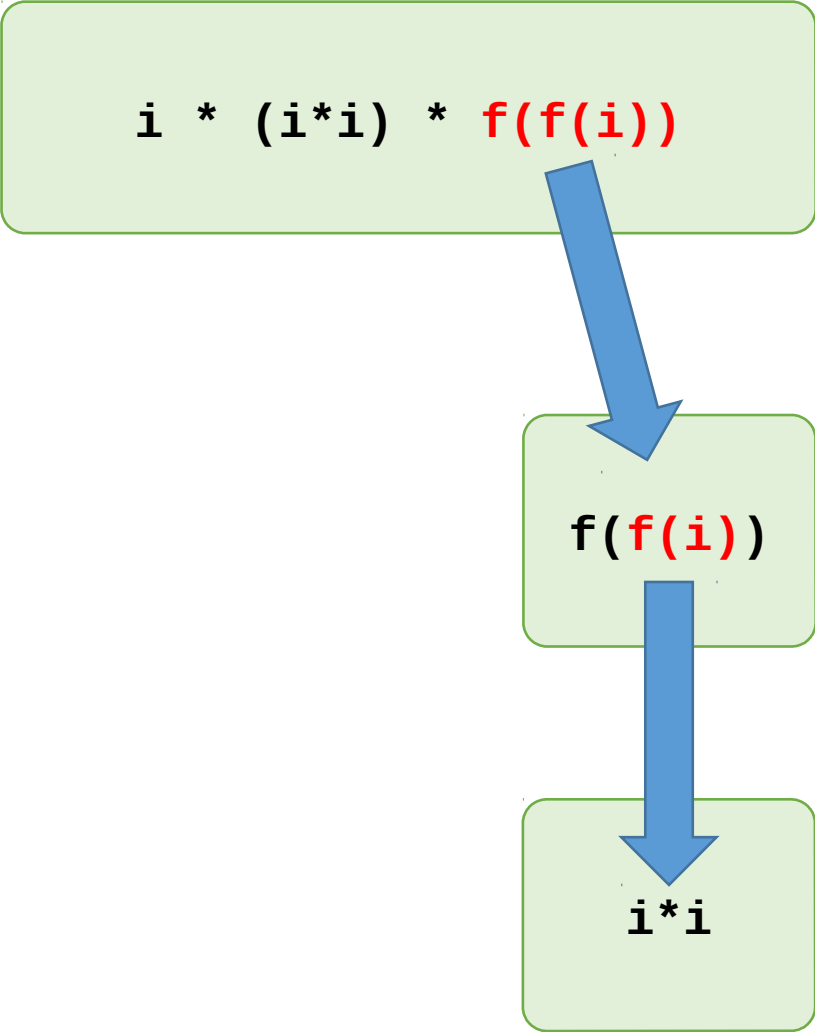
int g (int i) {
    return i * f(i) * f(f(i));
}

void h (int i) {
    std::cout << g(i) << "\n";
}

int main () {
    int i;
    std::cin >> i;
    h(i);
    return 0;
}
```

# Exercise 2

`i * (i*i) * f(f(i))`



`f(f(i))`

`i*i`

```
#include <iostream>

int f (int i) {
    return i * i;
}

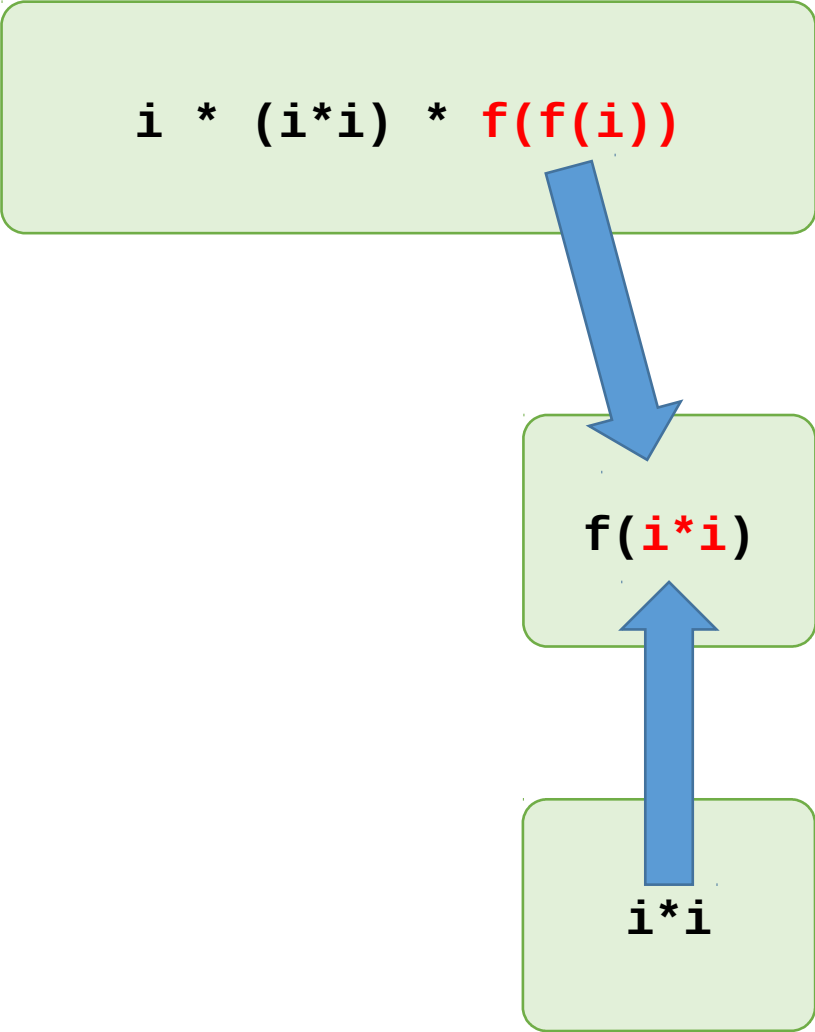
int g (int i) {
    return i * f(i) * f(f(i));
}

void h (int i) {
    std::cout << g(i) << "\n";
}

int main () {
    int i;
    std::cin >> i;
    h(i);
    return 0;
}
```

# Exercise 2

`i * (i*i) * f(f(i))`



`f(i*i)`

`i*i`

```
#include <iostream>

int f (int i) {
    return i * i;
}

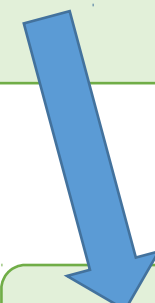
int g (int i) {
    return i * f(i) * f(f(i));
}

void h (int i) {
    std::cout << g(i) << "\n";
}

int main () {
    int i;
    std::cin >> i;
    h(i);
    return 0;
}
```

# Exercise 2

`i * (i*i) * f(f(i))`



`f(i*i)`

```
#include <iostream>

int f (int i) {
    return i * i;
}

int g (int i) {
    return i * f(i) * f(f(i));
}


void h (int i) {
    std::cout << g(i) << "\n";
}

int main () {
    int i;
    std::cin >> i;
    h(i);
    return 0;
}
```



# Exercise 2

`i * (i*i) * f(f(i))`



`(i*i)*(i*i)`

```
#include <iostream>

int f (int i) {
    return i * i;
}

int g (int i) {
    return i * f(i) * f(f(i));
}


void h (int i) {
    std::cout << g(i) << "\n";
}

int main () {
    int i;
    std::cin >> i;
    h(i);
    return 0;
}
```

# Exercise 2

`i * (i*i) * ((i*i)*(i*i))`

`(i*i)*(i*i)`



```
#include <iostream>

int f (int i) {
    return i * i;
}

int g (int i) {
    return i * f(i) * f(f(i));
}

void h (int i) {
    std::cout << g(i) << "\n";
}

int main () {
    int i;
    std::cin >> i;
    h(i);
    return 0;
}
```

# Exercise 2

```
i * (i*i) * ((i*i)*(i*i))
```

```
#include <iostream>

int f (int i) {
    return i * i;
}

int g (int i) {
    return i * f(i) * f(f(i));
}

void h (int i) {
    std::cout << g(i) << "\n";
}

int main () {
    int i;
    std::cin >> i;
    h(i);
    return 0;
}
```

# Exercise 2

```
i * (i*i) * ((i*i)*(i*i))
```

This is  
 $i^7$

```
#include <iostream>

int f (int i) {
    return i * i;
}

int g (int i) {
    return i * f(i) * f(f(i));
}

void h (int i) {
    std::cout << g(i) << "\n";
}

int main () {
    int i;
    std::cin >> i;
    h(i);
    return 0;
}
```