

# SOM2IF15 – Compilation

## AST Transformation

Frédéric Louergue



UNIVERSITE D'ORLEANS

2022

## With For Loop

```
int sum(int array table)
{
    int sum = 0
    foreach(int value : table)
        sum += value
    return sum
}

void printAll (char array array text)
{
    int n = length(text);
    for(int i = 0; i < n; i++)
        print (line [i])
}
```

## Without For Loop

```
int sum(int array table)
{
    int sum = 0
    int counter = 0
    while(counter < length(table)){
        int value = table[counter]
        sum += value
        counter += 1
    }
    return sum
}

void printAll (char array array text)
{
    int n = length(text)
    int i = 0
    while(i < n){
        print (text[i])
        i = i + 1
    }
}
```

# The Visitor VisitorCopy

- ▶ Performs a deep copy of an Ast
- ▶ Just by overriding some `visit` methods we can replace one Ast node by another
- ▶ It enables program transformations
- ▶ Easy if one node replaced by one node
- ▶ Slightly more difficult when one node needs to be replaced by several nodes

# Two Transformations

## Simplification of Assignments

- ▶ Before:  $x += 1$
- ▶ After:  $x = x + 1$

## Elimination of Assignment Expressions

- ▶ Before:  $y = 42 + (x++)$
- ▶ After:

$y = 42 + x$

$x = x + 1$