Réseaux et sécurité

Exercices – 01

Frédéric Loulergue

Université d'Orléans
Laboratoire d'Informatique Fondamentale d'Orléans

Fall 2022

Exercise 1

What are the problems?

```
/*@ ensures (a < b  && tmp == -1) ||
            (a == b && tmp == 0) ||
            (a > b  && tmp == 1); */
int compare(int a, int b)
{
  int tmp;
  if (a < b) tmp = - 1;
  if (a > b) tmp = 1;
  return tmp;
}
```

Exercise 2

Contract

Write a correct and complete functional contract for the `compare` function, using only ==> as a logical connective.

```c
int compare(int a, int b)
{
  int tmp = 0;
  if (a < b) tmp = - 1;
  if (a > b) tmp = 1;
  return tmp;
}
```

Exercise 3

Contract

Write a correct and complete functional contract that avoids runtime errors for the `incr`, `decr`, and `identity` functions.

```
int incr(int x){ return x + 1; }

int decr(int x){ return x - 1; }

int identity(int x){
  int tmp = decr(x);
  tmp = incr(tmp);
  return tmp;
}
```

Exercise 4

Necessary Condition

For a variable x of type **int**, what do you think about the following formulas as part of a precondition?

- INT_MIN <= x <= INT_MAX
- INT_MIN < x <= INT_MAX
- x > INT_MAX

Exercise 5

Contract

For dichotomic search, the following function is supposed to return the middle index in an array where `start` is the starting index and `end` the ending index.

Write a correct and complete functional contract that avoids overflows.

```
int mid(int start, int end)
{
  return (start + end) / 2;
}
```

Is there a way to rewrite the function so that there are more valid values for `start` and `end` without having oveflows?