

Software Certification using Frama-C (second part)

1. (a) Program a c function which computes the sum of integers from 0 to n without multiplication.
(b) Prove the code using the following specification :

```
/*@  
  requires 0 <= n <= 1000;  
  ensures 2*\result == n*(n+1);  
*/
```

2. (a) Program a c function which takes as argument an integer n and computes the sum of the first n odd numbers.
(b) Show that it verifies the following specification :

```
/*@  
  requires 0 <= n <= 1000;  
  ensures \result == n * n;  
  assigns \nothing;  
*/
```

3. (a) Program a c function which takes as arguments three arrays of the same size and modifies the third one such that it contains the sum of the first two arrays.
(b) Give a specification using the language ACSL.

```
/*@ requires ...;  
   @ ensures ...;  
   @*/  
void somme (int t[], int s[], int r[], int taille) { ... }
```

- (c) Prove that your program verifies the specification.
4. (a) Program a c function which computes the multiplication of two integers without using the multiplication.
(b) Specify the behaviour of your program using ACSL.

```
/*@ requires ...;  
   @ ensures ...;  
   @*/  
int mult(int x, int y) { ... }
```

- (c) Prove that the function verifies this specification.
5. (a) Program a c function to test if an array is sorted in increasing order.
(b) Specify the behaviour of your function using the language ACSL.

```
/*@ requires ...;  
   @ ensures ...;  
   @*/  
int trie(int t[], int taille) { ... }
```

- (c) Prove that the function verifies this specification.
6. (a) Program a C function which takes as arguments the arrays of integers and which modifies the third one to contain the concatenation of the first two..
- (b) Specify the behaviour of your function using ACSL.

```

/*@ requires ...;
   @ ensures ...;
   @*/
void concat (int t[], int tt, int s[], int ts, int r[])

```

- (c) Prove that the function verifies this specification.