Advanced C exam

EPITA – International Master

january 2016

2h – All documents allowed (no PC, no smartphone)

Please be as careful as you can when you write : it's sometimes just impossible for me to decrypt some sentences !

Applying a function to each element of an given array

You have to write a generic function named `apply` whose goal is to compute the result of another function when applied in turn to every element (that can be of any type) of an array. The results are stored in a second array.

More precisely, the function takes 5 parameters :

— The array containing the elements used as parameters.

— The number of elements.

— The size in memory of an element.

— The function to call on each element of the array.

— Finally, the result array containing the return values of the function calls.

For instance, say we have a `triple` function whose role is, as its name indicates, to compute the triple of the value passed as parameter. If we also have defined array `a` as containing integers `{1,4,2,8,5,7}`, then calling

```
apply(a, 6, sizeof(int), triple, b);
```

will result in array `b` containing the following values : `{3,12,6,24,15,21}`. We admit that the user of `apply` has previously created array `b`, which is of the same type and size as array `a`.

You will write three files :

— `apply.h` : declares the profile of the apply generic function ;

— `apply.c` : implements the apply generic function ;

— `testApply.c` : defines a small main test program illustrating the behaviour of the `apply` function in the context described above (array of integers and `triple` function)