

NOM : \_\_\_\_\_ Prénom : \_\_\_\_\_ Groupe : \_\_\_\_\_

1) Donnez le type, puis complétez le code Ocaml des fonctions suivantes :

*Recherche d'une valeur maximale dans un tableau « t » d'au moins une case*

```
a. let max t =
  let m = _____
  in
    for i = ___ to _____
    do
      _____
      _____
    done;
  _____;;
```

*Savoir si un tableau « t » d'au moins une case est trié croissant (utiliser uniquement « i »)*

```
b. let est_trie t =
  let n = Array.length t
  and i = ref _____
  in
    while (!i < _____ ) && ( _____ <= _____ )
    do
      i := !i + 1
    done;
  _____;;
```

2) Pour l'ensemble des expressions suivantes, vous donnerez pour chacune son type et éventuellement sa valeur.

```
a. let f1 n =
  let fc i = i + 1
  in Array.init (n + 1) fc;;
```

\_\_\_\_\_

\_\_\_\_\_

```
b. f1 3;;
```

\_\_\_\_\_

\_\_\_\_\_

```
c. let f2 n = Array.init n f1;;
```

\_\_\_\_\_

\_\_\_\_\_

```
d. f2 3;;
```

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

3) *Suite de Fibonacci*

On rappelle la définition de la suite de *Fibonacci* :  $F_0 = 0$ ,  $F_1 = 1$  et pour tout entier  $i > 1$ ,  $F_i = F_{i-1} + F_{i-2}$ . Réaliser (au verso) un fonction « fibo » qui prend en paramètre un entier « n » et retourne un tableau de « n » cases à valeurs entières contenant les n premières valeurs de la suite de Fibonacci.