Contrôle de IF3.01 du 23/10/2001

1. Écrivez une fonction `neg_bf` qui prend en argument une fonction `f: bool -> bool` et qui rend un boléen. La valeur de `neg_bf f` est `true` si `f` vaut `true` sur au moins une valeur booléenne, et `false` sinon.

```
# let neg_bf f = (f true)||(f false);;
val neg_bf : (bool -> bool) -> bool = <fun>
```

2. Écrivez une fonction de type `char -> int` qui associe aux cinq voyelles 'a', 'i', 'u', 'e', 'o' leur rang dans cet ordre, habituel pour la langue japonaise ('a' vaut 1, ..., 'o' vaut 5). La fonction doit faire de même avec les caractères majuscules, et rendre 0 pour tout autre caractères *alphabétique*, c'est-à-dire compris entre 'a' et 'z' ou entre 'A' et 'Z'.

```
# let jap c = match c with
          'a'|'A' -> 1
        | 'i'|'I' -> 2
        | 'u'|'U' -> 3
        | 'e'|'E' -> 4
        | 'o'|'O' -> 5
        | 'a'..'z'|'A'..'Z' -> 0
        | _ -> -1;;
val jap : char -> int = <fun>
```

3. Écrivez une fonction `frec` dont l'argument est de la forme `(n,(a,b))`, dont la valeur est indéfinie si `n` est négatif, et dont la valeur est la somme

$$\sum_{i=0}^{n}(x-i)(y+i)$$

si l'entier `n` est positif ou nul. Quel est le type de cette fonction ?

```
# let rec frec (n,(a,b)) = match n with
          n when (n<0) -> -1
        | 0 -> a*b
        | n-> (a-n)*(b+n) + frec (n-1,(a,b));;
val frec : int * (int * int) -> int = <fun>
```

4. Donenz le type des expressions suivantes :

   - `let f a (b,c) = if a then b+c else b-c;;`

     `val f : bool -> int * int -> int = <fun>`

- ```
  let x = ‘‘abcde’’ in if 2 mod 2 = 0 then x^x else ‘‘ ’’;;
  ```
  ```
  - : string = "abcdeabcde"
  ```
- ```
  let g x y = List.hd x + List.hd y;;
  ```
  ```
  val g : int list -> int list -> int = <fun>
  ```
- ```
  let h x = match x with (0,_) -> false | (n,y) -> n>y ;;
  ```
  ```
  val h : int * int -> bool = <fun>
  ```
  ```