

1.

```
let rec f x = function
  | 0 -> -1
  | n -> n + f (x+1) (n/3)
;;
```

 Quel est le type de cette fonction ?

- A int -> int -> int B int -> unit
- C (int * int) -> int D int -> int

2.

```
let rec f x = function
  | 0 -> -1
  | n -> n + f (x+1) (n/3)
;;
```

 Que renvoie f 0 4 ?

- A Entre dans une boucle infinie B -1
- C Une erreur D 4

3.

```
let rec f x = function
  | 0 -> -1
  | n -> n + f (x+1) (n/3)
;;
```

 Que renvoie f 0 ?

- A Une fonction de type int -> int B Un objet de type unit
- C Une erreur D -1

4.

```
let x y z = y = 4*z in x 3;;
```

 Que renvoie cette phrase ?

- A une fonction de type int -> bool B une erreur
- C une fonction de type int -> int D 12

5.

```
let rec f(x) = x * f(x-1);;
```

 Que renvoie f(4) ?

- A 4! B int fonction de type int -> int * int
- C Rien, entre dans une boucle infinie D 0

6.

```
let rec f(x,y) = match y with
  | 0 -> x
  | u -> f(x+1,u/2)
;;
```

 Cette fonction f est...

- A de type int -> int B de type int -> unit
- C de type int -> int -> int D de type int * int -> int



7.

```
let rec f(x,y) = match y with
  0 -> x
  | u -> f(x+1,u/2)
;;
```

 Cette fonction f est...

- A récurсивe terminale B on ne peut pas savoir
- C récurсивe non terminale

8.

```
let rec f(x,y) = match y with
  0 -> x
  | u -> f(x+1,u/2)
;;
```

 f(1,2) renvoie...

- A Rien (boucle infinie) B 0
- C 3 D 1
- E 2

9.

```
let rec f(x,y) = match y with
  0 -> x
  | u -> f(x+1,u/2)
;;
```

 f(x,y) renvoie toujours une valeur quel que soit les entiers relatifs x et y.

- A FAUX B VRAI

10.

```
let rec f(x,y) = match y with
  0 -> x
  | u -> f(x+1,u-1)
;;
```

 f(x,y) renvoie toujours une valeur quel que soit les entiers relatifs x et y.

- A VRAI B FAUX



Corrigé

1.a

2.d

3.a

4.a

5.c

6.c

7.a

8.c

9.b

10.b



