



Questionnaire du QCM d'INF201, INF231

Répondre sur la feuille réponse à part

Question 1 ♣ Quelle est le type de l'expression `(int_of_float 2.3)`

☐ A unit

☐ C float

☐ B int

☐ D bool

Question 2 ♣ Quel est le type de la fonction `f` définie par `let f (x:int) : int = x/2`

☐ A float

☐ C `int→int→int`

☐ B `int→int`

☐ D int

Question 3 ♣ L'expression `2=3=false`

☐ A s'évalue a `true`

☐ B s'évalue a `false`

☐ C renvoie une erreur de type

Question 4 ♣ L'expression `if a then not a else false` est équivalente à

☐ A ne peut pas s'exprimer simplement avec des opérateurs logiques.

☐ B `not a`

☐ C `false`

Question 5 ♣ L'expression `if a then b else a` est équivalente à

☐ A `a || b`

☐ F `not (not a || not b)`

☐ B `(not a) && b`

☐ G ne peut pas s'exprimer simplement avec des opérateurs logiques.

☐ C `a && b`

☐ D `not (a && b)`

☐ H `(not a) || b`

☐ E `not (a || b)`

Question 6 ♣ La fonction `f` définie par `let f a x = if a then x = 2 else x = 3`

☐ A assigne un `bool` à la variable `x`

☐ E prend deux entiers en arguments

☐ B est de type `bool→int→ bool`

☐ F prends un `bool` et un `int` en arguments

☐ C renvoie un `bool`

☐ G assigne un `int` à la variable `x`

☐ D renvoie un `int`

Question 7 ♣ Le type `intervalle` défini par `type intervalle = float*float` est

☐ A un type énuméré

☐ D un type synonyme du type `float*float`

☐ B un drôle de type

☐ E un type somme

☐ C un type produit



Question 8 ♣ On veut modéliser les faces d'un dé à six faces. On peut pour cela définir le type face par

- ☐ A `type face = Un|Deux|Trois|Quatre|Cinq|Six;;`
- ☐ B `type face = 1|2|3|4|5|6;;`
- ☐ C `type face = int;; (*restreint au entier de 1 à 6*)`
- ☐ D `type face = Un*Deux*Trois*Quatre*Cinq*Six;;`
- ☐ E `type face (x:int)= 1<=x && x<=12;;`

Question 9 ♣ Si x et y n'ont pas encore été définis, la valeur de l'expression `let x = 10 in let x = 3 in let y = x+4 in x+y;;` est

- ☐ A 7
- ☐ B 3
- ☐ C 10
- ☐ D 17
- ☐ E 24
- ☐ F inexistante (problème de "let in" imbriqués)

Question 10 ♣ Si x et y n'ont pas encore été définis, la valeur de l'expression `let x = 10 and y = 3+x in x+y;;` est

- ☐ A 7
- ☐ B inexistante (erreur).
- ☐ C 23
- ☐ D 24
- ☐ E 10

Question 11 ♣ On définit la fonction `monOu` par `let monOu (x :bool) (y :bool):bool = x || y ;;`
Lorsqu'on évalue les expressions suivantes lesquelles donnent une erreur (division par zéro).

- ☐ A `monOu (0 = 2/0) true`
- ☐ B `(0 = 2/0) || false`
- ☐ C `monOu true (0 = 2/0)`
- ☐ D `true || (0 = 2/0)`
- ☐ E `false || (0 = 2/0)`
- ☐ F `(0 = 2/0) || true`

Question 12 ♣ On définit la fonction `f` par `let f b x = if (b && x>0) then x+1 else x-1;;`

- ☐ A `(f true 0)` vaut 1
- ☐ B `(f true 0)` vaut -1
- ☐ C `b` est de type bool
- ☐ D `(f true 4)` vaut 3
- ☐ E `x` est de type int
- ☐ F `(f true 4)` vaut 5

Question 13 ♣ Soient `f` et `g` définies par `let f x = 2*x and g y = y-4;;`. Quelle est la valeur de `g (f (f (g 9)))`

- ☐ A 42
- ☐ B 23
- ☐ C 31
- ☐ D 2020
- ☐ E 16
- ☐ F 63

Question 14 ♣ On définit la fonction `inclus` par `let inclus (bi1,bs1 :int*int) (bi2,bs2 :int*int) : bool = (bi2<bi1 && bs1<bs2)`
Parmi les expressions suivantes lesquelles valent `true`.

- ☐ A `inclus (4,5) (2,3)`
- ☐ B `inclus (2,5) (4,8)`
- ☐ C `inclus (4,5) (2,8)`
- ☐ D `inclus (3,5) (2,6)`
- ☐ E `inclus (2,3) (4,5)`
- ☐ F `inclus (8,3) (7,5)`



Question 15 ♣ On implémente les triplets d'entiers par le type `triplet` comme suit:

```
type triplet=int*int*int;;
```

On cherche à définir une fonction `third` qui à un triplet d'entier, renvoie son troisième élément. Cocher les cases des implémentations correctes (s'il y en a).

- | | |
|--|--|
| <input type="checkbox"/> A <code>let third (t:triplet):int=t(3);;</code> | <input type="checkbox"/> E <code>let third ((a,b,c):triplet):int=c;;</code> |
| <input type="checkbox"/> B <code>let third (t:triplet):int=t.[3];;</code> | <input type="checkbox"/> F <code>let third (t:triplet):int=t[3];;</code> |
| <input type="checkbox"/> C <code>let third (t:triplet):int= let (a,b,c)=t in c;;</code> | <input type="checkbox"/> G <code>let third (t:triplet):int= match t with (a,b,c)->c;;</code> |
| <input type="checkbox"/> D <code>let third (t:triplet):int=t.(3);;</code> | |

Question 16 ♣ On soumet le code suivant à l'interprète `ocaml`:

```
type pfc=Pierre|Feuille|Ciseau;;
```

```
let perd (x:pfc) (y:pfc) : bool =
```

```
  match x,y with  
  |Pierre,Feuille->true  
  |Ciseau,Pierre->true  
  |Feuille,Ciseau->true;;
```

- | | |
|--|--|
| <input type="checkbox"/> A Il accepte la définition de la fonction <code>perd</code> sans warning | <input type="checkbox"/> D Il émet un warning pour la définition de la fonction <code>perd</code> : "this pattern matching is not exhaustive" |
| <input type="checkbox"/> B l'expression <code>(perd Feuille Pierre)</code> s'évalue à <code>true</code> | <input type="checkbox"/> E l'expression <code>(perd Feuille Pierre)</code> renvoie une erreur |
| <input type="checkbox"/> C Il dit qu'il y a une erreur lors de la définition de la fonction <code>perd</code> | <input type="checkbox"/> F l'expression <code>(perd Feuille Pierre)</code> s'évalue à <code>false</code> |

Question 17 ♣ On définit la fonction `f` par

```
let f (x:int) (y:int) : int=  
  if (x<y) then x else y;;
```

L'expression `(f x y)` vaut

- | | |
|--|--|
| <input type="checkbox"/> A le minimum de <code>x</code> et <code>y</code> | <input type="checkbox"/> D <code>false</code> si $y \geq x$ |
| <input type="checkbox"/> B <code>true</code> si $y < x$ | <input type="checkbox"/> E <code>true</code> si $x < y$ |
| <input type="checkbox"/> C le maximum de <code>x</code> et <code>y</code> | <input type="checkbox"/> F <code>false</code> si $x \geq y$ |

Question 18 ♣ On définit la fonction `f` par

```
let f (x1:float) (x2:float) (x3:float) (x4:float): float=
```

```
  let xmax=(max (max x1 x2) (max x1 x4)) and xmin=(min (min x1 x2) (min x1 x4)) in  
  (x1+.x2+.x3+.x4-.xmin-.xmax)/.2.;;
```

L'expression `(f 3. 1. 5. 11.)` vaut

- | | | | | | | |
|---------------------------------------|---------------------------------------|---------------------------------------|---------------------------------------|--------------------------------------|---------------------------------------|---------------------------------------|
| <input type="checkbox"/> A 2.5 | <input type="checkbox"/> B 6.5 | <input type="checkbox"/> C 10. | <input type="checkbox"/> D 4.5 | <input type="checkbox"/> E 4. | <input type="checkbox"/> F 3.5 | <input type="checkbox"/> G 7.5 |
|---------------------------------------|---------------------------------------|---------------------------------------|---------------------------------------|--------------------------------------|---------------------------------------|---------------------------------------|



Question 19 ♣ On définit les majuscules, minuscules et chiffres à l'aide des trois types suivants:

```
type minus = char (*restreint aux minuscules*)
```

```
type majus = char (*restreint aux majuscules*)
```

```
type digit = char (*restreint aux chiffres*)
```

On souhaite maintenant définir un type `alphanum` qui regroupe les caractères des trois sortes.

Quels sont les solutions possibles:

☐ A `type alphanum =majus*minus*chiffre`

☐ B `type alphanum = char (*restreint aux minuscules, majuscules et chiffres*)`

☐ C `type alphanum =
 maj(majus)
 |min(minus)
 |chiffre(digit)`

☐ D `type alphanum =
 maj of majus
 |min of minus
 |chiffre of digit`

☐ E `type alphanum =
 Maj of majus
 |Min of minus
 |Chiffre of digit`

Question 20 ♣ On soumet le code suivant à l'interprète ocaml:

```
let f (x:bool) (y:bool) : bool =  
  match x,y with  
  |true, _ ->true  
  |a , _ ->not a  
  |true , true ->false  
  |false , false->false;;
```

☐ A `f x y` vaut toujours `true`

☐ B `f x y` vaut toujours `not x`

☐ C `f x y` vaut toujours `not y`

☐ D L'interprète dit qu'il y a une erreur

☐ E `f x y` vaut toujours `x||y`

☐ F Aucune des autres réponses n'est juste.

☐ G `f x y` vaut toujours `x && y`

☐ H `f x y` vaut toujours `x || ((not x)||y)`

☐ I `f x y` vaut toujours `x`



Feuille Réponse du QCM d'INF201, INF231

A rendre avec votre copie

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 |
| 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 |
| 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 |
| 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 |

← codez votre numéro d'étudiant ci-contre, et écrivez votre nom et prénom ci-dessous.

Nom et prénom :

.....

Veillez à bien noircir les cases au stylo (noir de préférence). Toutes les questions valent 2 points chacune et ont une bonne réponse ou plus. Le score obtenu pour chaque question ne peut pas être négatif. Une case noircie alors qu'elle ne devrait pas annule les points de la questions. Si telle n'est pas le cas le score de la question est le nombre de cases bien noircie sur le nombre de cases qu'il faut noircir.

Veillez à bien lire toutes les questions. Elles ne sont pas classées par ordre de difficulté croissante.

Q1 : ☐ A ☐ B ☐ C ☐ D

Q2 : ☐ A ☐ B ☐ C ☐ D

Q3 : ☐ A ☐ B ☐ C

Q4 : ☐ A ☐ B ☐ C

Q5 : ☐ A ☐ B ☐ C ☐ D ☐ E ☐ F ☐ G ☐ H

Q6 : ☐ A ☐ B ☐ C ☐ D ☐ E ☐ F ☐ G

Q7 : ☐ A ☐ B ☐ C ☐ D ☐ E

Q8 : ☐ A ☐ B ☐ C ☐ D ☐ E

Q9 : ☐ A ☐ B ☐ C ☐ D ☐ E ☐ F

Q10 : ☐ A ☐ B ☐ C ☐ D ☐ E

Q11 : ☐ A ☐ B ☐ C ☐ D ☐ E ☐ F

Q12 : ☐ A ☐ B ☐ C ☐ D ☐ E ☐ F

Q13 : ☐ A ☐ B ☐ C ☐ D ☐ E ☐ F

Q14 : ☐ A ☐ B ☐ C ☐ D ☐ E ☐ F

Q15 : ☐ A ☐ B ☐ C ☐ D ☐ E ☐ F ☐ G

Q16 : ☐ A ☐ B ☐ C ☐ D ☐ E ☐ F

Q17 : ☐ A ☐ B ☐ C ☐ D ☐ E ☐ F

Q18 : ☐ A ☐ B ☐ C ☐ D ☐ E ☐ F ☐ G

Q19 : ☐ A ☐ B ☐ C ☐ D ☐ E

Q20 : ☐ A ☐ B ☐ C ☐ D ☐ E ☐ F ☐ G ☐ H ☐ I