



# Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>QCM Semaine 1</b>	<b>3</b>
2.1	algorithmique . . . . .	3
2.2	language C . . . . .	4
	<b>Les réponses aux questionnaires</b>	<b>5</b>

Page d'accueil

Page de Titre

Sommaire

◀◀ ▶▶

◀ ▶

Page 1 de 4

Retour

Plein Ecran

Fermer

Quitter



## 1. Introduction

Pour chaque semaine, un QCM d'algorithmique est suivi de QCM pour le langage informatique étudié.

- Pour commencer un QCM on doit cliquer sur le bouton **Début**.
- Pour le terminer, on clique sur **Fin**.
- Le score s'affiche.

Vous pouvez recommencer le QCM en cliquant de nouveau sur Début ou voir les bonnes réponses en cliquant sur **Réponses**.

Un clic sur les cases **vertes** permet de voir une petite explication sur la bonne réponse.

Quand une réponse littérale est à fournir et que l'on a demandé la correction, un clic sur **Ans** fournit la réponse dans la case en bas du questionnaire. Un appui sur SIFT+clic donne la réponse complète si elle existe.

Page d'accueil

Page de Titre

Sommaire



Page 2 de 4

Retour

Plein Ecran

Fermer

Quitter

## 2. QCM Semaine 1

### 2.1. algorithmique

Répondre aux 7 questions.

1. Une constante  
peut être lue au clavier n'est pas modifiable est modifiable par affectation
2. Un bon programmeur  
ne commente pas son programme, il sait ce qu'il fait ne commente pas son programme, cela fait du texte inutile commente son programme et le documente
3. Une variable doit être initialisée  
Vrai Faux
4. Les types fondamentaux sont au nombre de  
4 5 6 7
5. A quel niveau l'information du type d'une variable est-elle la plus utile ?  
écriture du programme compilation exécution
6. Non (A Ou B) = Non A Et Non B  
Faux Vrai
7. L'expression suivante  $3 + 5 * 12 - 5$   
91 56 58
8. Un entier n a trois valeurs possibles (-1 0 1) et l'on souhaite exécuter 3 séquences respectives (séquence 1, séquence 2 et séquence 3) suivant la valeur de cet entier. On utilisera de préférence :  
Si .. Alors .. SinonSi .. Alors .. SinonSi .. Alors .. FinSi  
Si .. Alors .. Sinon ..Si .. Alors .. Sinon .. FinSi .. FinSi



Page d'accueil

Page de Titre

Sommaire

◀ ▶

◀ ▶

Page 3 de 4

Retour

Plein Ecran

Fermer

Quitter

## 2.2. langage C

Répondre aux 3 questions.

1. Pour déclarer une variable x de type réel, on écrit :

Float x;                    x: double;                    double x;                    x: Réel;                    Réel x;

2. Pour définir une constante appelé MAX de valeur 10, on écrit :

#define MAX 5;                    #define MAX=5                    #define MAX 5                    #define int MAX=5;                    int MAX=5;

3. Pour écrire une expression booléenne qui indique si une variable entière v est différente de 0, on écrit :

v /= 0                    v == 1                    v <> 0                    v != 0                    v <- 0

4. À l'issue de la déclaration `int n = 10 / 4;`, la valeur de la variable n est :

0                    0.4                    2                    2.5                    6

5. Pour initialiser une variable entière n (`int n`) au clavier, on fait :

scanf("%i", &n);                    scanf("%d", n);                    scanf("%d", &n);                    printf("%d", n);                    scanf("n=%i", &n);

6. Pour exécuter une séquence d'instructions si un entier a est égal à zéro, on écrit :

<code>if (a == 0) {</code> Séquence <code>}</code>	<code>if a == 0 {</code> Séquence <code>}</code>	<code>if (a = 0) {</code> Séquence <code>}</code>	<code>if (a = 0)</code> Séquence
--	--	---	-------------------------------------

7. Quelles instructions permutent les valeurs de a et b

<code>a = temp</code> <code>b = a</code> <code>temp = b</code>	<code>temp = a</code> <code>a = b</code> <code>b = temp</code>	<code>temp = a</code> <code>b = temp</code> <code>a = b</code>
--	--	--

8. Quel est la valeur de somme après cette boucle ?

```
int i;  
int somme;  
for (i = 1; i <= 10; i++) {  
    somme = somme + i;  
}
```



Page d'accueil

Page de Titre

Sommaire

◀ ▶

◀ ▶

Page 4 de 4

Retour

Plein Ecran

Fermer

Quitter

## Les réponses aux questionnaires

**Réponse :** Une constante est comme son nom l'indique non modifiable.

[Retour au questionnaire.](#)



[Page d'accueil](#)

[Page de Titre](#)

[Sommaire](#)



[Page 5 de 4](#)

[Retour](#)

[Plein Ecran](#)

[Fermer](#)

[Quitter](#)

**Réponse :** Il n'y a que les mauvais programmeurs qui ne commentent pas leur programme.

[Retour au questionnaire.](#)



[Page d'accueil](#)

[Page de Titre](#)

[Sommaire](#)



*Page 6 de 4*

[Retour](#)

[Plein Ecran](#)

[Fermer](#)

[Quitter](#)

**Réponse :** Une variable doit être initialisée avant toute utilisation, sinon sa valeur est quelconque.

[Retour au questionnaire.](#)



[Page d'accueil](#)

[Page de Titre](#)

[Sommaire](#)



*Page 7 de 4*

[Retour](#)

[Plein Ecran](#)

[Fermer](#)

[Quitter](#)

**Réponse :** Les types fondamentaux sont au nombre de cinq ( Entier, Réel, Booléen, Caractère, Chaîne de caractères).

[Retour au questionnaire.](#)



[Page d'accueil](#)

[Page de Titre](#)

[Sommaire](#)



[Page 8 de 4](#)

[Retour](#)

[Plein Ecran](#)

[Fermer](#)

[Quitter](#)



**Réponse :** Le compilateur est capable de détecter les erreurs de compatibilité de type dans une expression. Notons que certains environnements de programmation peuvent exploiter cette information à l'exécution pour vérifier que la valeur d'une variable est compatible avec son type, par exemple lors de la saisie. [Retour au questionnaire.](#)



*Page d'accueil*

*Page de Titre*

*Sommaire*



*Page 9 de 4*

*Retour*

*Plein Ecran*

*Fermer*

*Quitter*

Réponse : Cette expression est vrai.

[Retour au questionnaire.](#)



[Page d'accueil](#)

[Page de Titre](#)

[Sommaire](#)



[Page 10 de 4](#)

[Retour](#)

[Plein Ecran](#)

[Fermer](#)

[Quitter](#)

**Réponse :** La multiplication est prioritaire sur l'addition donc le calcul s'effectue  $5 \times 12 : 60$  puis  $3 + 60 : 63$  puis  $63 - 5 : 58$ .

[Retour au questionnaire.](#)



[Page d'accueil](#)

[Page de Titre](#)

[Sommaire](#)



Page 11 de 4

[Retour](#)

[Plein Ecran](#)

[Fermer](#)

[Quitter](#)

**Réponse :** Les deux solutions peuvent convenir, la première étant plus logique. N'oublions pas que Selon .. Dans .. .. FinSelon aurait été mieux dans ce cas ou un entier ne peut avoir que 3 valeurs distinctes.

[Retour au questionnaire.](#)



[Page d'accueil](#)

[Page de Titre](#)

[Sommaire](#)



Page 12 de 4

[Retour](#)

[Plein Ecran](#)

[Fermer](#)

[Quitter](#)

**Réponse :** En C, le type Float n'existe pas. C'est le type **float**, pour les réels en simple précision et **double** pour les réels en double précision. Réel est le type algorithmique. Il n'existe pas non plus en C.

La syntaxe en C pour déclarer une variable correspond aux réponses 1, 3 et 4. Les deux autres correspondent à la syntaxe algorithmique. La bonne solution est donc la 3 : **double x**;

Notons que quand on a le choix entre un réel double et simple précision, il est préférable de prendre un réel double précision, en particulier les bibliothèques standard manipulent les réels double précision.

[Retour au questionnaire.](#)



*Page d'accueil*

*Page de Titre*

*Sommaire*



*Page 13 de 4*

*Retour*

*Plein Ecran*

*Fermer*

*Quitter*

**Réponse :** Pour définir une constante en C, on utilise le préprocesseur. Il s'agit de définir la valeur d'une macro (ici MAX). Il n'y a donc pas de signe égal (=). Il ne faut surtout pas mettre de point-virgule (;) à la fin : la valeur de MAX et 5 et non 5; !

La dernière réponse ne définit pas une constante mais déclare une variable !

En conséquence, la bonne réponse est **#define** MAX 5.

[Retour au questionnaire.](#)



[Page d'accueil](#)

[Page de Titre](#)

[Sommaire](#)



[Page 14 de 4](#)

[Retour](#)

[Plein Ecran](#)

[Fermer](#)

[Quitter](#)

**Réponse :** La différence se note  $\neq$  en C. Les réponses 1 et 3 sont des notations utilisées dans d'autres langages (Ada et Pascal). La réponse 2 est fausse : ce n'est pas parce que  $v$  est différent de 0 qu'il est égale à 1.  $v == 1$  est une condition suffisante mais pas nécessaire !

Enfin la dernière réponse correspond à l'affectation et non à test !

[Retour au questionnaire.](#)



*Page d'accueil*

*Page de Titre*

*Sommaire*



*Page 15 de 4*

*Retour*

*Plein Ecran*

*Fermer*

*Quitter*

**Réponse :** Il s'agit d'une division entière puisque les opérands 10 et 4 sont entières. En conséquence, le résultat est 2. Le reste de la division entière  $10 \% 4$  donnerait 2 également :  $10 = 2 * 4 + 2$ .

[Retour au questionnaire.](#)



*Page d'accueil*

*Page de Titre*

*Sommaire*



*Page 16 de 4*

*Retour*

*Plein Ecran*

*Fermer*

*Quitter*



**Réponse :** La réponse 4 est fausse puisque printf permet d'afficher mais pas de saisir. C'est bien scanf qu'il faut utiliser.  
La réponse 2 est fausse car le paramètre de scanf doit être l'adresse de la variable, là où la valeur lue doit être écrite.  
La réponse 5 est fausse car le format impose à l'utilisateur de taper les caractères n = devant l'entier à lire.  
Les réponses 1 et 3 sont toutes les deux possibles.

[Retour au questionnaire.](#)



*Page d'accueil*

*Page de Titre*

*Sommaire*



*Page 17 de 4*

*Retour*

*Plein Ecran*

*Fermer*

*Quitter*

**Réponse :** En C, il faut nécessairement mettre les parenthèses autour de la condition booléenne (c'est parce qu'il n'y a pas d'équivalent du **then**).

D'autre part, il ne faut pas confondre == qui réalise un test d'égalité et = qui correspond à l'affectation.

Enfin, notons que si les accolades ne sont pas obligatoires en C, il faut nécessairement les mettre dès que la séquence est composée d'au moins deux instructions.

[Retour au questionnaire.](#)



[Page d'accueil](#)

[Page de Titre](#)

[Sommaire](#)



[Page 18 de 4](#)

[Retour](#)

[Plein Ecran](#)

[Fermer](#)

[Quitter](#)



**Réponse :** Il suffit d'exécuter les différents programmes pour voir le quel fonctionne. Il faut bien sûr prendre des valeurs différentes pour a et b, par exemple a == 3 et b == 5. On prend la valeur indéterminée pour tmp, tmp = ?.

L'exécution du premier programme donne a == ? et b == ?.

L'exécution du second programme donne a == 5 et b == 3. Le bon résultat.

L'exécution du troisième programme donne a == 3 et a == 3.

Seule la deuxième exécution donne le bon résultat. Les exécutions réalisées avec 3 et 5 peuvent être généralisés à toutes valeurs possibles de a et b (par exemple va et vb). Le bon résultat est donc le deuxième programme.

```
temp = a
a = b
b = temp
```

Vérifier le sens pour l'affectation si vous vous êtes trompé et la manière d'exécuter en séquence les instructions.

[Retour au questionnaire.](#)

Page d'accueil

Page de Titre

Sommaire



Page 19 de 4

Retour

Plein Ecran

Fermer

Quitter

**Réponse :** La variable somme n'étant pas initialisée, elle est indéterminée et l'instruction  $\text{somme} = \text{somme} + i$  conserve une valeur indéterminée au résultat.

Si vous exécutez ce programme, il y a bien une valeur qui sera affichée mais elle ne dépend pas du programmeur mais du contexte d'exécution.

Ce programme est faux car il est interdit d'utiliser une valeur indéterminée!

Si la variable somme avait été initialisée à 0, alors le résultat aurait été 55!

[Retour au questionnaire.](#)



*Page d'accueil*

*Page de Titre*

*Sommaire*



*Page 20 de 4*

*Retour*

*Plein Ecran*

*Fermer*

*Quitter*