



# BARHOUMI Nizar (barhoumi.nizar.issat@gmail.com)

Campagne : Développeur PHP Full Stack - Intermédiaire

Langage(s) de programmation : Javascript, PHP, SQL

Langage : Français

Date : 13/02/2019

SCORE

**94%**

2 070 / 2 200 pts

RANG

**1**

/ 19

DURÉE

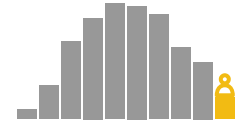
**0H33**

/ 0H54

MEILLEUR QUE

**98%**

des développeurs



## Javascript



**95%**

(910 / 960)

Connaissance du langage



**100%**

(387 / 387)

Fiabilité



**73%**

(133 / 183)

Modélisation



**100%**

(40 / 40)

Résolution de problèmes



**100%**

(350 / 350)



## PHP



**88%**

(560 / 640)

Connaissance du langage



**82%**

(360 / 440)

Résolution de problèmes



**100%**

(200 / 200)



## SQL



**100%**

(600 / 600)

Connaissance du langage



**100%**

(600 / 600)

## Question 1: [JS] Concat



Javascript



00:17 / 00:45



40 / 40 pts

### ? Question

Que devez-vous ajouter après les deux lignes suivantes pour que le tableau A contienne une concaténation du tableau A et B ?

```
var a = [1,2,3];  
var b = [4,5,6];
```

### 📝 Réponse

- a = a + b;
- a.concat(b);
- b.concat(a);
- a = a.concat(b);

### > Résultat



Réponse correcte

Connaissance du langage +40pts

## Question 2: [CSS] Couleurs



Javascript



00:13 / 00:25



40 / 40 pts



### Question

En CSS, dans quel format est spécifiée la couleur #A1B2C3 ?

---



### Réponse

- Cyan Magenta Jaune Noir hexadécimal
  - Rouge Vert Bleu décimal
  - Rouge Vert Bleu hexadécimal
  - Cyan Magenta Jaune Noir octal
- 



### Résultat



Réponse correcte

Connaissance du langage +40pts

## Question 3: [JS] Simple expression booléenne



Javascript



01:32 / 02:00



100 / 100 pts

### ? Question

$a(i, j)$  devrait retourner *true* si au moins un des arguments est égal à 1 ou si leur somme est égale à 1.

Les deux arguments donnés seront toujours des nombres.

Par exemple :

$a(1, 5)$  retourne *true*

$a(2, 3)$  retourne *false*

$a(-3, 4)$  retourne *true*



### Réponse

```
1 // JavaScript code below
2 // Use printErr(...) to debug your solution.
3
4 function a(i, j) {
5   // Your code goes here
6   if(i == 1 || j == 1 || i+j == 1){return true;}else{return false;}
7 }
```



### Résultat



Retourne true si i ou j est égal à 1, sinon false

Connaissance du langage +67pts



Retourne true si i+j est égal à 1

Fiabilité +33pts

## Question 4: [JS] Correction



Javascript



01:55 / 05:00



200 / 200 pts

### ? Question

La fonction *factorial* suivante, écrite par votre collègue Frédéric est cassée. Réparez la afin qu'elle renvoie la factorielle du nombre donné en paramètre.

Rappel :  $factorial(n) = 1 * 2 * 3 * \dots * n$

### ✎ Réponse

```
1 // JavaScript code below
2 // Use printErr(...) to debug your solution.
3
4 function factorial(n) {
5     if (n == 0) {
6         return 1;
7     }
8     else {
9         return n * factorial(n-1);
10    }
11 }
```

### > Résultat



La fonction factorial fonctionne  
Connaissance du langage +200pts

## Question 5: [CSS] Ordre d'expression



Javascript



00:31 / 00:40



40 / 40 pts

### ? Question

En CSS, à quoi correspondent les valeurs ci-dessous ?

*margin: 10px 20px 15px -10px;*

---

### 📝 Réponse

- top bottom left right
  - top left bottom right
  - top right bottom left**
  - face front reverse alpha
  - Cette règle CSS est invalide
- 

### > Résultat



Réponse correcte

Connaissance du langage +40pts

## Question 6: [JS] Threads



Javascript



00:12 / 00:25



40 / 40 pts



### Question

JavaScript est multi-threadé.

---



### Réponse



Vrai



Faux

---



### Résultat



Réponse correcte  
Modélisation +40pts

## Question 7: [JS] Range Sum



Javascript



02:28 / 05:00



200 / 200 pts

### ? Question

Écrivez le corps de la fonction `calc(array, n1, n2)`.

`array` est un tableau d'entiers. Les paramètres `n1` et `n2` sont des entiers définis par la relation  $0 \leq n1 \leq n2 < \text{array.length}$ .

La fonction `calc` doit retourner la somme des entiers de `array` dont l'index appartient à l'intervalle `[n1; n2]`.

### 📝 Réponse

```
1 // Use printErr(...) to debug your solution.
2
3 /**
4  * Returns the sum of integers whose index is between n1 and n2.
5  * @returns {Number}
6  */
7 function calc(array, n1, n2) {
8     var s=0;
9     for(i=n1;i<=n2;i++){
10         s=s+array[i];
11     }
12     return s;
13 }
```

### > Résultat



Tests basiques

Résolution de problèmes +100pts



Tests avancés

Résolution de problèmes +100pts



## Question 8: [JS] Combinaisons possibles dans un tournoi



Javascript



07:59 / 15:00



250 / 300 pts

### ? Question

Vous organisez un tournoi d'échecs dans lequel les joueurs s'affronteront en duel.

Pour former les duels on procède ainsi : d'abord on tire au hasard un joueur, ensuite on tire au hasard son opposant parmi les joueurs restants. Cette paire forme un des duels du tournoi. On procède de la même manière pour former toutes les paires.

Dans cet exercice, on souhaiterait connaître **le nombre de paires** qu'il est possible d'obtenir sachant que l'ordre des opposants dans une paire n'a pas d'importance.

Par exemple, avec 4 joueurs nommés A, B, C et D, il est possible d'obtenir 6 paires différentes : AB, AC, AD, BC, BD, CD.

Implémentez `count()` pour retourner le nombre de paires possibles. Le paramètre `n` correspond au nombre de joueurs.

Essayez d'optimiser votre solution pour que, dans l'idéal, la durée de traitement soit la même quel que soit `n`.

Données :  $2 \leq n \leq 30000$

---

## Réponse

```
1 // JavaScript code below
2 // Use printErr(...) to debug your solution.
3 /*function factoriel(n){
4     if (n==0){
5         return 1;
6     }
7     else{
8         return n * factoriel(n-1);
9     }
10 }*/
11
12 function count(n) {
13     //combinaison = factoriel(n)/( factoriel(n-2) * 2 );
14     var s = 0;
15     for(i=0;i<=n-1;i++){
16         s = s+i;
17     }
18     return s;
19 }
```

---

## Résultat

- La solution fonctionne avec un jeu de données simple  
Résolution de problèmes +150pts
- La solution retourne le bon résultat en un temps constant  
Fiabilité +50pts
- La solution fonctionne avec des données proches de 10000  
Fiabilité +100pts

## Question 9: Sous-chaîne



PHP



00:16 / 00:35



40 / 40 pts

### ? Question

Saisissez le nom de la fonction qui sert à retourner une partie d'une chaîne de caractères à partir d'un index de début et d'une longueur.

### ✎ Réponse

substr

### > Résultat



Réponse correcte

Connaissance du langage +40pts

### ✓ Réponse(s) correcte(s)

- substr
- substr()
- substr ()

## Question 10: Opérateur sur les bits : >>



PHP



00:20 / 00:20



0 / 40 pts

⚠ Le temps alloué à cette question s'est écoulé. La réponse du candidat a été automatiquement récupérée à la fin du décompte.

### ? Question

Quel est le résultat de l'opération  $2 \gg 1$  ?

### ✎ Réponse

- 0
- 1
- 2
- 3
- 4

### > Résultat

⊗ Réponse incorrecte  
Connaissance du langage +40pts

## Question 11: define()



PHP



00:10 / 00:25



40 / 40 pts

### ? Question

Quelle déclaration est une constante valide en PHP ?

---

### ✎ Réponse

- `define("MAXSIZE", 100);`
  - `#define MAXSIZE 100`
  - `static final MAXSIZE = 100;`
  - `constant MAXSIZE = 100;`
- 

### > Résultat



Réponse correcte  
Connaissance du langage +40pts

## Question 12: Exit



PHP



00:19 / 00:30



40 / 40 pts

### ? Question

Saisissez le nom de la structure de langage, alias de la fonction `exit()`, qui est couramment utilisée dans les scripts.

### ✎ Réponse

die()

### > Résultat



Réponse correcte

Connaissance du langage +40pts

### ✓ Réponse(s) correcte(s)

- die
- die()
- die();

## Question 13: Range Sum



PHP



02:53 / 05:00



200 / 200 pts

### ? Question

Écrivez le corps de la fonction `calc($array, $n1, $n2)`.

`$array` est un tableau d'entiers. Les paramètres `$n1` et `$n2` sont des entiers définis par la relation  $0 \leq n1 \leq n2 < \text{count}(\$array)$ .

La fonction `calc` doit retourner la somme des entiers de `$array` dont l'index appartient à l'intervalle `[$n1; $n2]`.

### 📝 Réponse

```
1 <?php
2 /**
3  * @return integer the sum of integers whose index is between n1 and n2
4  */
5 function calc($array, $n1, $n2) {
6     $s=0;
7     for($i=$n1;$i<=$n2;$i++){
8         $s += $array[$i];
9     }
10    return $s;
11 }
12 ?>
```

### > Résultat



Tests basiques

Résolution de problèmes +100pts



Tests avancés

Résolution de problèmes +100pts

## Question 14: SQL - NOT NULL



SQL

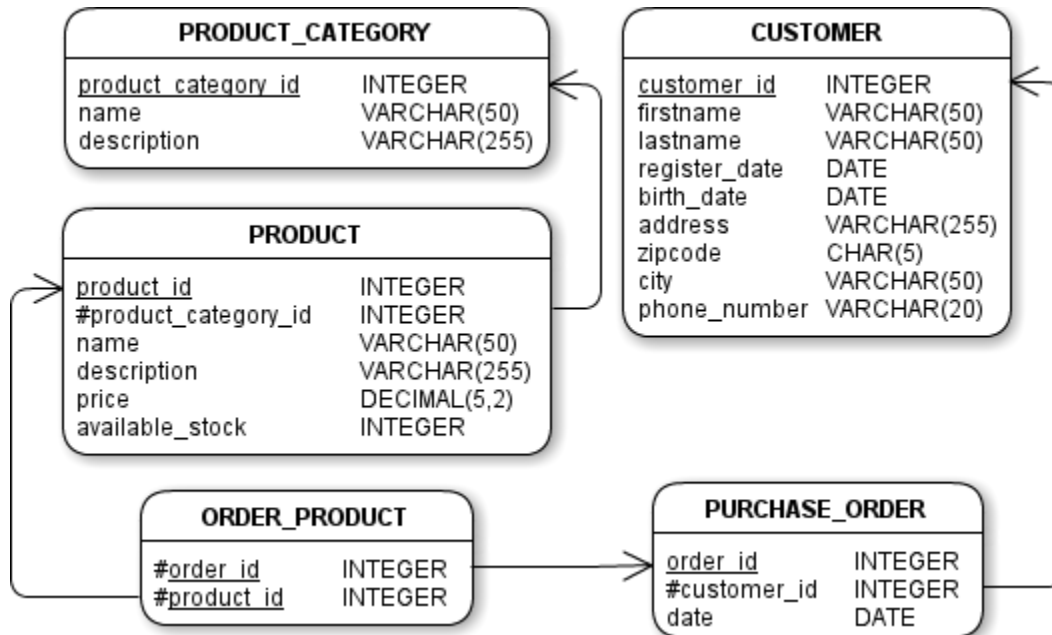


04:17 / 05:00



200 / 200 pts

### ? Question



Modifier la requête pour selectionner uniquement les clients ayant un code postal (zipcode) égal à 75000 ou 34000 et ayant une date de naissance (birth\_date) définie.

N'afficher que colonnes LASTNAME et FIRSTNAME dans cet ordre.

Exemple de sortie :

LASTNAME	FIRSTNAME
RIVER	John
JACKSON	Elizabeth
PUTTER	James



## Réponse

```
1 -- SQL request(s) below
2 SELECT lastname,firstname
3 FROM customer
4 where (zipcode = '75000' or zipcode = '34000') and birth_date is not null;
```

---

## Résultat



Utilisation de "not null"  
Connaissance du langage +200pts

## Question 15: SQL - LIKE



SQL

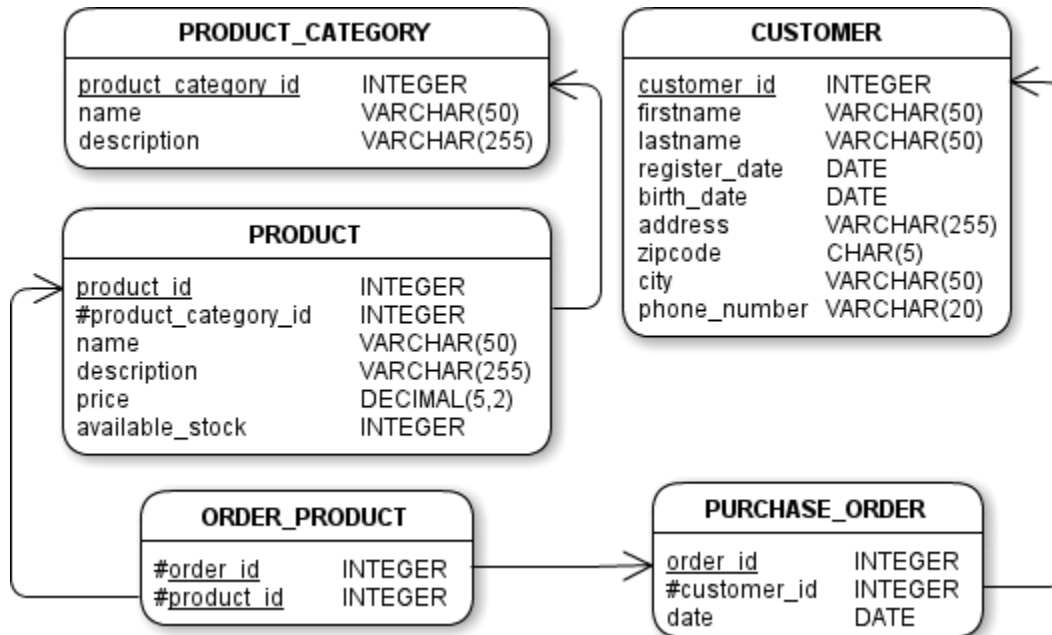


04:31 / 05:00



200 / 200 pts

### ? Question



Modifier la requête pour sélectionner uniquement les noms et prénoms des clients (customer) dont le nom commence par la lettre 'W', triés par ordre alphabétique de noms puis de prénoms.

N'afficher que les colonnes LASTNAME et FIRSTNAME dans cet ordre.

Exemple de sortie :


```
-----  
| LASTNAME | FIRSTNAME |  
-----  
| WHITTARD | Bill      |  
| WILLIAMS | John     |  
-----
```

## Réponse

```
1 -- SQL request(s) below
2 SELECT lastname,firstname
3 FROM customer
4 where lastname like 'W%'
5 order by lastname , firstname ;
```

---

## Résultat

 Utilisation de "like", "order by"  
Connaissance du langage +200pts

## Question 16: SQL - NOT EXISTS



SQL

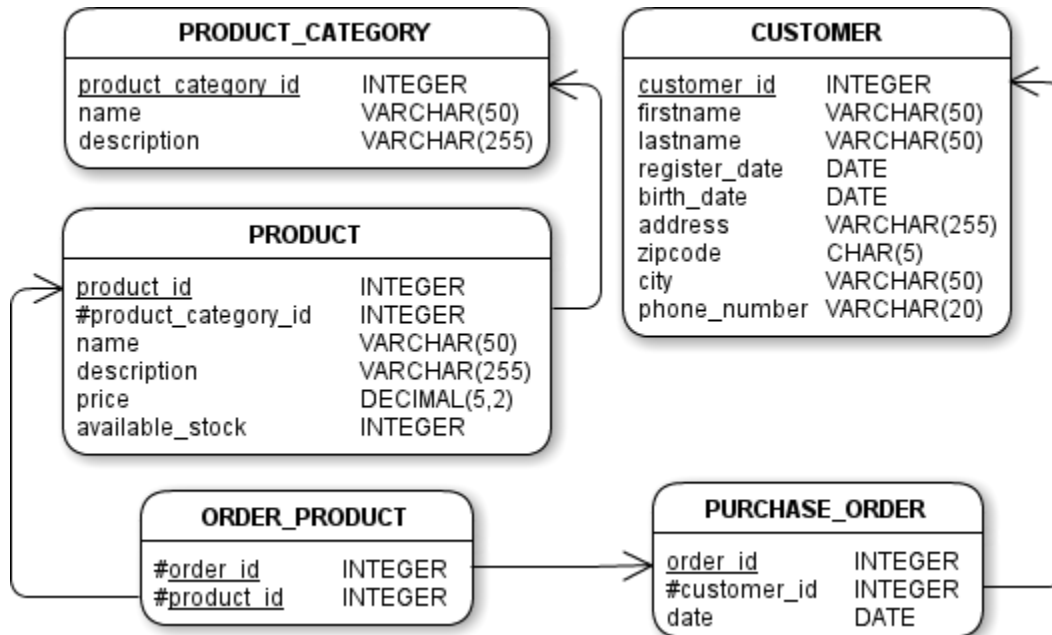


02:15 / 05:00



200 / 200 pts

### ? Question



Modifier la requête pour selectionner uniquement les clients (customer) n'ayant aucun ordre achat (purchase\_order) associé.

Afficher uniquement la colonne CUSTOMER\_ID.


Exemple de sortie :

CUSTOMER_ID
8
5




## Réponse

```
1 -- SQL request(s) below
2 SELECT c.customer_id
3 FROM customer c
4 where c.customer_id not in (select p.customer_id from purchase_order p )
```

## Résultat

 Utilisation de "not exists" ou "not in"  
Connaissance du langage +200pts

### Question 17: Convention de nommage des variables

 PHP  00:15 / 00:20  40 / 40 pts


## Question

Par convention, un nom de variable d'objet commence par une majuscule.

## Réponse

- Vrai  
 Faux

## Résultat

 Réponse correcte  
Connaissance du langage +40pts

## Question 18: Opérateur sur les bits : &



PHP



00:18 / 00:20



40 / 40 pts

### ? Question

En base 2 (binaire), combien font  $0001 \& 0001$  ?

---

### ✎ Réponse

0010

0000

0001

---

### > Résultat



Réponse correcte

Connaissance du langage +40pts

## Question 19: Taille d'une chaîne



PHP



00:20 / 00:20



0 / 40 pts

⚠ Le temps alloué à cette question s'est écoulé. La réponse du candidat a été automatiquement récupérée à la fin du décompte.

---

### ? Question

Quelle fonction retourne le nombre de caractères d'une chaîne ASCII ?

---

### ✎ Réponse

- length
  - size
  - strlen
  - len
- 

### > Résultat

⊗ Réponse incorrecte  
Connaissance du langage +40pts

## Question 20: Opérateur sur les bits : |




PHP



00:25 / 00:25



40 / 40 pts

 Le temps alloué à cette question s'est écoulé. La réponse du candidat a été automatiquement récupérée à la fin du décompte.

---

### Question

En base 2 (binaire), combien font **01** | **11** ?

---

### Réponse

- 00
  - 01
  - 10
  - 11
  - 100
- 

### Résultat



Réponse correcte

Connaissance du langage +40pts



## Question 21: Evaluer une chaîne de caractères



PHP



00:22 / 00:30



40 / 40 pts

### Question

Saisissez le nom de la fonction qui évalue une chaîne de caractères comme un script PHP.

---

### Réponse

eval()

---

### Résultat



Réponse correcte

Connaissance du langage +40pts

---

### Réponse(s) correcte(s)

- eval
- eval()
- eval ()

## Question 22: Type d'une variable



PHP



00:12 / 00:30



40 / 40 pts

### ? Question

Saisissez le nom de la fonction qui permet d'afficher le type et le contenu d'une variable.

### ✎ Réponse

var\_dump()

### > Résultat



Réponse correcte

Connaissance du langage +40pts

### ✓ Réponse(s) correcte(s)

- var\_dump
- var\_dump()
- var\_dump ()
- print\_r
- print\_r()
- print\_r ()
- var\_export
- var\_export()
- var\_export ()

## Question 23: Opérateur unitaire (i++)



PHP



00:22 / 00:25



40 / 40 pts

### ? Question

```
$i = 0;  
echo $i++;
```

Saisissez le résultat qu'affiche ce bout de code.

---

### ✎ Réponse

0

---

### > Résultat



Réponse correcte

Connaissance du langage +40pts

---

### ✓ Réponse(s) correcte(s)

- 0

# Glossaire

## Connaissance du langage

La mesure de cette compétence permet de déterminer l'expérience du candidat dans la pratique d'un langage de programmation. **Privilégiez cette compétence si, par exemple, vous recherchez un développeur qui devra être rapidement opérationnel.**

## Design

Cette mesure fournit une indication sur la capacité du candidat à appliquer des solutions standard pour résoudre des problèmes récurrents. Un développeur ayant un bon niveau dans cette compétence augmentera la qualité (maintenabilité, évolutivité) de vos applications. Cette compétence ne dépend pas spécifiquement d'une technologie. **Privilégiez cette compétence si, par exemple, vous recherchez un développeur qui sera amené à travailler sur les briques qui structurent vos applications, à anticiper les besoins de demain pour développer des solutions pérennes.**

## Résolution de problèmes

Cette compétence correspond aux aptitudes du candidat à comprendre et à structurer son raisonnement pour trouver des solutions à des problèmes complexes. Cette compétence ne dépend pas spécifiquement d'une technologie. **Privilégiez cette compétence si, par exemple, vos applications ont une composante technique importante (R&D, innovation).**

## Fiabilité

La fiabilité caractérise la capacité du candidat à réaliser des solutions qui prennent en compte les cas particuliers. Plus cette compétence est élevée, plus vos applications sont robustes (moins de bugs).