

Apprenez à programmer en C !

⌚ 40 heures 📶 Moyenne

Licence CC BY NC SA

Mis à jour le 28/06/2021



Découvrez les bases de la programmation en C

Bravo ! Vous avez réussi cet exercice !

Compétences évaluées

- ✓ Utiliser un environnement de développement
- ✓ Créer des algorithmes simples en C

Question 1

Comment s'appelle le programme chargé de traduire un code source en binaire ?

- ☐ Le diffuseur
- ✓ ☒ Le compilateur
- ☐ Le binarisateur

Le compilateur est un élément indispensable dans les langages de programmation. Sans lui, on ne pourrait pas traduire le code source dans un langage binaire compréhensible par l'ordinateur !

Question 2

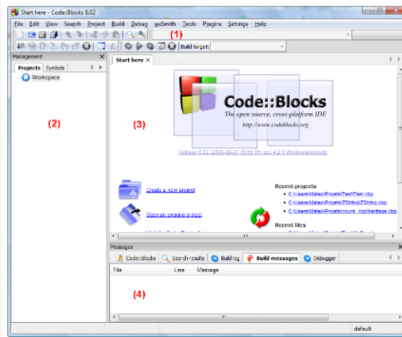
Lequel de ces programmes n'est pas un IDE ?

- ✓ ☒ Bloc-Notes
- ☐ Microsoft Visual Studio
- ☐ Code::Blocks
- ☐ Xcode

Bien qu'on puisse en théorie écrire du code source sous Bloc-Notes, ce n'est qu'un éditeur et pas un IDE. Un IDE comprend un éditeur, un compilateur et un débogueur, ainsi que de nombreuses autres fonctionnalités destinées à faciliter le développement !

Question 3

Où s'afficheront les erreurs de compilation dans Code::Blocks d'après la capture suivante ?



- ☐ Zone (1)
- ☐ Zone (2)
- ☐ Zone (3)
- ✓ ☒ Zone (4)

La zone en bas de Code::Blocks affiche traditionnellement les messages d'erreur. Surveillez-la attentivement, car vous risquez d'y passer du temps !

Question 4

Que signifie le `return 0` dans le code suivant ?

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    printf("Hello world!\n");
    return 0;
}
```

Les bases de la programmation en C

1. Vous avez dit programmer ?

2. Ayez les bons outils !

3. Votre premier programme

4. Un monde de variables

5. Une bête de calcul

6. Les conditions

7. Les boucles

8. TP : Plus ou Moins, votre premier jeu

9. Les fonctions

10. Entraînez-vous à améliorer le jeu du "plus ou moins"

Quiz : Découvrez les bases de la programmation en C

- ☐ Le programme a effectué 0 actions
- ☐ Le programme ne s'est pas bien exécuté
- ✓ ☒ Le programme s'est bien exécuté

Le code 0 signifie que tout s'est bien passé. Tout autre code signifierait qu'une erreur a eu lieu.

Question 5

Quel symbole permet d'effectuer un retour à la ligne à l'écran ?

- ☐ \t
- ✓ ☒ \n
- ☐ Il suffit d'appuyer sur la touche Entrée, voyons !

Le symbole \n signifie "retour à la ligne". Appuyer sur "Entrée" risque au contraire de casser votre code !

Question 6

Si ma variable "compteEnBanque" est un long qui vaut 6 500 000 (soyons fous), qu'est-ce que cette ligne de code affichera à l'écran ?

```
printf("Vous avez %ld euros sur votre compte", compteEnBanque);
```

- ☐ Vous avez %ld euros sur votre compte
- ✓ ☒ Vous avez 6 500 000 euros sur votre compte
- ☐ Vous avez ld euros sur votre compte, compteEnBanque

Le symbole %ld sera remplacé par la valeur de la variable qui suit, soit 6 500 000 ici !

Question 7

Quelle est la seule mémoire qui n'est pas vidée lorsque l'ordinateur est éteint ?

- ☐ Registres
- ☐ Mémoire cache
- ☐ Mémoire vive
- ✓ ☒ Disque dur

Toutes les mémoires ont besoin d'être alimentées en électricité pour pouvoir retenir les nombres. Toutes ? Non, car une irréductible mémoire parvient à conserver ses informations même sans électricité : c'est le disque dur. Grâce à lui, votre PC peut se souvenir qui il est à chaque démarrage !

Question 8

Combien vaudra la variable "resultat" après cette opération ?

```
resultat = (8 / 3) - 2;
```

- ☐ -2
- ✓ ☒ 0
- ☐ 1
- ☐ 2

En C, la division (8/3) ne retient que la partie entière. Répondez donc à la question "Combien de fois 3 est-il entièrement présent dans 8 ?" : 2 fois. Donc (8/3) = 2. Si ensuite vous retirez 2, cela fait 2 - 2 = ... 0 !

Question 9

Quel est le problème de ce switch ?

```
switch (variable)
{
    case 5:
        printf("Salut");
    case 12:
        printf("Bonjour");
    default:
        printf("Au revoir");
}
```

- ✓ ☒ Il manque des instructions break
- ☐ Il faut mettre un point-virgule à la fin du switch
- ☐ Il faut ouvrir des accolades pour chaque "case"
- ☐ C'est "case default" et pas "default" tout court

Sans break, le switch exécutera aussi toutes les actions des "case" qui suivent !

Question 10

Laquelle de ces boucles for pourrait afficher les messages suivants dans la console ?

Ligne n°1

Ligne n°3

Ligne n°5

Ligne n°7

- ✓ ☒ `for (compteur = 1 ; compteur < 9 ; compteur += 2)`
- ☐ `for (compteur = 1 ; compteur <= 7 ; compteur++)`
- ☐ `for (compteur = 0 ; compteur < 9 ; compteur += 2)`
- ☐ `for (compteur = 1 ; compteur < 8 ; compteur++)`

La première ligne affiche 1, donc cela signifie que notre variable compteur a été initialisée à 1. On peut déjà éliminer la troisième solution. Ensuite, on voit que notre variable augmente de 2 en 2 : on a donc forcément utilisé un `compteur += 2`. Il ne reste donc plus que la première solution de possible, et ça tombe bien parce que c'est la bonne ! 😊

Question 11

Combien de fois le message "Salut" sera-t-il affiché ici ?

```
int compteur = 14;
```

```
while (compteur < 15)
{
    printf("Salut\n");
}
```

- ☐ 0 fois
- ☐ 1 fois
- ☐ 14 fois
- ☐ 15 fois

✓ ☒ C'est une boucle infinie

compteur vaut 14 au départ. Comme c'est inférieur à 15, on rentre dans la boucle. On affiche "Salut", puis on repart au début de la boucle et on refait un test... Et comme compteur vaut toujours 14 (on n'a pas changé sa valeur !) la boucle va se répéter ainsi à l'infini ! C'est typiquement le genre d'erreur qu'il ne faut pas faire. N'oubliez pas de faire l'incréméntation, car si votre variable vaut toujours la même valeur, la condition sera toujours vraie ! 😊

Question 12

Dans quel cas l'instruction return n'est pas obligatoire ?

- ☐ Quand la fonction ne prend aucun paramètre en entrée
- ✓ ☒ Quand la fonction est de type void
- ☐ Quand la fonction doit renvoyer 0

Une fonction de type void ne renvoie rien, donc pas besoin de mettre de return 😊

Question 13

Que sont les paramètres d'une fonction ?

- ☐ Des indications sur le nom de la fonction
- ☐ Des indications sur la valeur qu'elle doit renvoyer
- ✓ ☒ Des variables qu'on lui envoie pour qu'elle puisse travailler

Les paramètres d'une fonction sont des variables qu'on envoie à cette fonction. Le rôle de la fonction est de faire des calculs à partir de ces paramètres et de renvoyer un résultat. Notez qu'une fonction peut ne pas prendre de paramètres (mais c'est rare), et elle n'est pas obligée de renvoyer un résultat non plus, elle peut très bien seulement afficher quelque chose à l'écran par exemple.

Question 14

Quel est le problème de cette fonction qui est censée calculer le carré de la variable nombre ?

Rappel : le carré d'un nombre, c'est nombre * nombre (le nombre multiplié par lui-même)

```
int carre(int nombre){
    int resultat = 0;
    resultat = nombre * nombre;
}
```

- ✓ ☒ La fonction ne retourne aucune valeur
- ☐ La fonction ne marche pas car on a oublié un point-virgule quelque part
- ☐ Quel problème ? Il n'y a pas de problème !

Une fonction qui n'est pas de type `void` DOIT retourner une valeur. Il manque donc un `return` à la fin de la fonction.

Question 15

Combien de fonctions peut comporter un programme ?

- ☐ Une seule, le *main*
- ☐ Maximum 100

○ Maximum 1024

✓ ● Il n'y a pas de limite !

Vous pouvez créer autant de fonctions que nécessaire, ne vous limitez pas en pensant que cela va faire planter le programme ou le ralentir !

◀ **ENTRAÎNEZ-VOUS À AMÉLIORER LE JEU
DU "PLUS OU MOINS"**

LA PROGRAMMATION MODULAIRE ▶

Les professeurs



Mathieu Nebra

Entrepreneur à plein temps, auteur à plein temps et co-fondateur d'OpenClassrooms :o)



Ranga Gonnage

Développeur logiciel, mentor et enseignant.

Découvrez aussi ce cours en...



Livre



PDF

OPENCLASSROOMS

Qui sommes-nous ?

Alternance

Financements

Expérience de formation

Forum

Blog [↗](#)

Presse [↗](#)

OPPORTUNITÉS

Nous rejoindre [↗](#)

Devenir mentor [↗](#)

Devenir coach carrière [↗](#)

AIDE



FAQ

POUR LES ENTREPRISES

Former et recruter

EN PLUS

Boutique [↗](#)

Conditions générales d'utilisation

Politique de Protection des données
personnelles

Cookies

Accessibilité

🌐 Français [▼](#)

