

NOM : _____ PRENOM : _____

Pour chaque question se peut aussi qu'il n'y ait aucune, 1, 2, 3 ou 4 réponses de correctes. Cochez une ou plusieurs cases dans une même colonne.

En cas d'erreur, utilisez la colonne suivante.

Nota Bene :

Syntaxiquement fausse ⇔ la syntaxe du langage C n'est pas respectée dans son acception la plus courante. La réponse "est syntaxiquement fausse" exclue obligatoirement toutes les autres réponses.

1/ L'instruction : `scanf("%d", x) ;`

- | | | | |
|--|--------------------------|--------------------------|--------------------------|
| a. permet de saisir une variable pointée par x | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| b. affiche un message d'invite à l'écran | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| c. contient un champ | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| d. est syntaxiquement fausse | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

2/ L'instruction : `printf("Entrez la valeur de a : %d", a) ;`

- | | | | |
|--|--------------------------|--------------------------|--------------------------|
| a. permet de saisir une variable appelée a | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| b. affiche un message d'invite à l'écran | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| c. a doit être une variable de type int | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| d. est syntaxiquement fausse | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

3/ L'instruction : `#include{stdio.h}`

- | | | | |
|---|--------------------------|--------------------------|--------------------------|
| a. est indispensable à tout programme | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| b. inclus une bibliothèque à un programme | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| c. doit être placée en début de programme | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| d. est syntaxiquement fausse | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

4/ L'instruction : `for (k=0 ; k<10 ; k += 3) { instruc_suivantes; }`

- | | | | |
|--|--------------------------|--------------------------|--------------------------|
| a. fait que k prend les valeurs 0, 3, 6, 9 | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| b. permet de répéter les instruc_suivantes | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| c. est syntaxiquement fausse | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| d. impose que k soit entier | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

5/ L'instruction : `printf("a = %7.2d", 3.141592);`

- | | | | |
|------------------------------|--------------------------|--------------------------|--------------------------|
| a. affiche : a = __3.14 | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| b. affiche : a = 3.141592 | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| c. est syntaxiquement fausse | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| d. comporte un champ | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

6/ `'\n'` et `'\t'`

- | | | | |
|---|--------------------------|--------------------------|--------------------------|
| a. <code>'\t'</code> provoque un passage à la ligne | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| b. <code>'\t'</code> et <code>'\n'</code> sont des caractères de contrôle | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| c. <code>'\t'</code> insère des espaces | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| d. Aucun des choix ci-dessus | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

7/ L'instruction : `scanf("%d", &X) ;`

- | | | | |
|--|--------------------------|--------------------------|--------------------------|
| a. impose de X soit de type int * | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| b. accède à l'adresse de la variable X | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| c. provoque la saisie d'une variable | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| d. nécessite la bibliothèque stdlib.h | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

8/ En langage C

- | | | | |
|---|--------------------------|--------------------------|--------------------------|
| a. les variables sont toutes typées | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| b. les variables ont toutes une adresse | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| c. la valeur par défaut des variables est 0 | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| d. les variables doivent être déclarées | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

9/ L'instruction : `if (a) printf("ok\n"); else { instructions_2 ; }`

- | | | | |
|---|--------------------------|--------------------------|--------------------------|
| a. exécute toujours les <i>instructions_2</i> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| b. affiche "ok" si a vaut -3 | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| c. provoque une erreur si a est de type float | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| d. est syntaxiquement fausse | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

10/ le code suivant : `do { instructions ; } while (err < 1e-3) ;`

- | | | | |
|---|--------------------------|--------------------------|--------------------------|
| a. répète les <i>instructions</i> tant que err est supérieur à 1e-3 | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| b. on quitte la boucle si eps = -3 | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| c. impose que err soit modifié dans les instructions | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| d. les <i>instructions</i> sont exécutées au moins une fois | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

11/ Le code : `do{ int i = 0; printf("%d ; ",i); i++; } while(i<= 3);`

- | | | | |
|------------------------------------|--------------------------|--------------------------|--------------------------|
| a. affiche à l'écran 0 ; 1 ; 2 | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| b. affiche à l'écran 0 ; 1 ; 2 ; 3 | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| c. affiche n'importe quoi a priori | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| d. i est une variable entière | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

12/ Le code : `a=2; if (a = 3) printf("ok\n"); printf("bad\n");`

- | | | | |
|-------------------------------------|--------------------------|--------------------------|--------------------------|
| a. affiche « ok » | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| b. est syntaxiquement fausse | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| c. affiche « bad » | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| d. afficherait « ok » si a valait 3 | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

13/ Le code suivant : `float a[10];`

- | | | | |
|---|--------------------------|--------------------------|--------------------------|
| a. crée une variable dimensionnée appelée a | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| b. crée une variable entière a[9] | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| c. crée une variable a[10] | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| d. est syntaxiquement fausse | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

14/ Le code suivant : `int a[5] = { ,,1, 2 } ;`

- | | | | |
|---|--------------------------|--------------------------|--------------------------|
| a. crée une variable dimensionnée entière | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| b. initialise a[2] à 1 | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| c. initialise a[4] à 0 | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| d. est syntaxiquement fausse | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

15/ Le code suivant : `switch(c){ case 'a' : printf("a\n") ;`

`case 'A' :
default : printf("b\n") ; }`

- | | | | |
|------------------------------------|--------------------------|--------------------------|--------------------------|
| a. affiche « a » si c contient 'A' | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| b. impose que c soit de type char | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| c. est syntaxiquement fausse | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| d. affiche « b » dans tous les cas | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

16/ Après le code suivant : `int a[10][15] ;`

- | | | | |
|--|--------------------------|--------------------------|--------------------------|
| a. la variable <code>a[10][4]</code> existe | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| b. la variable <code>a[9]</code> existe : son type est int * | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| c. la variable <code>a[11]</code> existe | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| d. a peut être considérée comme une matrice carrée | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

17/ L'affirmation : «Lors de l'écriture d'une fonction il est conseillé l'écriture d'un prototype, d'un en tête et d'un return»

- | | | | |
|--|--------------------------|--------------------------|--------------------------|
| a. est exacte | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| b. est fausse : le return n'est pas conseillé | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| c. est fausse : l'en tête n'est pas conseillé | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| d. est fausse : le prototype n'est pas conseillé | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

18/ La fonction : `void calcul(float &x , int *y) { x = x+2*y; }`

- | | | | |
|---|--------------------------|--------------------------|--------------------------|
| a. est syntaxiquement fausse | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| b. nécessite deux paramètres | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| c. doit être appelée ainsi : <code>calcul(4.5, &A) ;</code> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| d. peut être appelée ainsi : <code>calcul(4.5, A) ;</code> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

19/ La fonction : `float truc(float a); { int k ; for (k=0 ; k<3 ; k++) a = 1/a; return a ; };`

- | | | | |
|---|--------------------------|--------------------------|--------------------------|
| a. est syntaxiquement fausse | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| b. comporte une variable locale | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| c. retourne 3 si on l'appelle par : <code>truc(3)</code> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| d. peut être appelée par : <code>printf("%d",truc(2));</code> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

20/ L'instruction : `typedef vect float[10] ;`

- | | | | |
|--|--------------------------|--------------------------|--------------------------|
| a. crée un nouveau type de variable appelé vect | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| b. est syntaxiquement fausse | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| c. crée une variable vect | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| d. est nécessaire à la fonction : <code>int F(vect a, int n)</code> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

21/ La commande : `F = fopen("toto.txt","R") ;`

- | | | | |
|---|--------------------------|--------------------------|--------------------------|
| a. ouvre le fichier toto.txt en lecture | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| b. est syntaxiquement fausse | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| c. impose que F soit de type FILE * | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| d. met TRUE dans F si le fichier existe | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

22/ L'instruction : `p = (char *)malloc(101*sizeof(char)) ;`

- | | | | |
|---|--------------------------|--------------------------|--------------------------|
| a. impose que p soit de type char * | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| b. nécessite la bibliothèque <code>stdlib.h</code> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| c. crée une variable dimensionnée | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| d. grâce à elle on peut accéder à <code>p[100]</code> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

23/ L'affirmation : "Une variable dimensionnée est un pointeur"

- | | | | |
|---|--------------------------|--------------------------|--------------------------|
| a. est fausse | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| b. utilise la syntaxe du pointeur parce que c'est une adresse | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| c. est exacte | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| d. En fait j'adore le chocolat ! | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

24/ Cochez les affirmations exactes dans la liste ci-dessous

- | | | | |
|--|--------------------------|--------------------------|--------------------------|
| a. un pointeur ne peut pas contenir l'adresse d'un tableau | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| b. un pointeur ne peut pas être transtypé | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| c. un pointeur ça ne sert à rien dans main ! | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| d. un pointeur est une variable comme une autre : il a son adresse | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

25/ Cochez les affirmations exactes dans la liste ci-dessous

- | | | | |
|--|--------------------------|--------------------------|--------------------------|
| a. c'est l'OS qui accède aux fichiers | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| b. il faut utiliser un pointeur pour gérer les fichiers | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| c. il n'existe que des fichiers texte | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| d. un fichier texte ne peut pas contenir des nombres réels | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

26 / Si l'on écrit : `float *P; int **K; K = &P;`

- | | | | |
|---------------------------------|--------------------------|--------------------------|--------------------------|
| a. P est un pointeur | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| b. K est un pointeur | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| c. *K est un pointeur, c'est P. | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| d. *P est un entier. | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |