

Banque d'exercices

Les fonctions

1. Quel est le résultat du programme

```
int f(int x) {  
    x = 10;  
    return x;  
}  
main(){  
    int y=2;  
    y=f(2);  
    printf("%d",y);  
}
```

2. Quel est le résultat du programme

```
void f(int x) {  
    x = 10;  
}  
main(){  
    int x=2;  
    f(x);  
    printf("%d",x);  
}
```

3. Quel est le résultat du programme

```
double f(int x, int y) {  
    return (x+y/2.0);  
}  
main(){  
    double y;  
    y=f(2,4);  
    printf("%.2lf",y);  
}
```

4. Quel est le résultat du programme

```
int f(int x) {  
    return (x==0);  
}  
main(){  
    int y;  
    y=f(2);  
    printf("%d",y);  
}
```

5. Quel est le résultat du programme

```
int f(int x) {
    int y ;
    if (x==0)
        y=1;
    else
        y=0;
    return y;
}
main(){
    int y;
    y=f(2);
    printf("%d",y);
}
```

6. Ecrire le code du main qui appelle la fonction addition et affiche le résultat. Tester avec les valeurs 2 et 3.

```
int addition (int a, int b){
    return (a+b);
}
```

7. Ecrire le code du main qui appelle la fonction addition et affiche le résultat. Tester avec toutes les valeurs de 0 à 9 (boucle for) pour a et 1 pour b.

8. Ecrire le code du main qui appelle 3 fois la fonction isLettreMin avec 'b', '0' et 'A' et affiche si c'est une lettre minuscule ou pas.

```
int isLettreMin(char c){
    return (c>='a' && c<='z');
}
```

9. Ecrire le code du main qui appelle la fonction isLettreMaj avec les lettres 'a', 'A' et '0' et affiche si c'est une lettre majuscule ou pas.

```
int isLettreMaj(char c){
    if(c>='A' && c<='Z')
        return 1;
    else
        return 0;
}
```

10. Ecrire le code du main qui appelle la fonction nombre avec les lettres '0', '4' et 'A' et '0' et affiche la valeur renvoyées par le fonction.

```
int nombre (char c){
    if(c>='0' && c<='9')
        return (c-'0');
    else
        return -1;
}
```

11. Ecrire le code du main qui teste la fonction affLignes pour 2, 4, 6, 8 et 10 (boucle for) après chaque appel de affLignes ajouter un saut de ligne dans la boucle for.

```
void affLignes(int n){
    for(int i=0;i<n;i++)
        printf("*");
}
```

Résultat attendu :

```
**
****
*****
*****
*****
```

12. Ecrire le code de AffCarre qui affiche un carre de taille n. Cette fonction appelle affLigne. Tester cette fonction affCarre dans le main pour les valeurs 2 et 8.

13. Ecrire le code de test (main) de la fonction mini avec x qui va de -10 à +10 (boucle for) et y =0.

```
double mini(double x, double y){
    if(x>y)
        return y;
    else
        return x;
}
```

14. Ecrire le code du main qui appelle la fonction moyenne et affiche le résultat. Tester avec toutes les valeurs de 0 à 9 (boucle for) pour a et 1 pour b.

```
double moyenne (int a, int b){
    return (a+b)/2.0;
}
```

15. Ecrire le code de la fonction division qui renvoie le résultat de a/b

```
double division(double a, double b);
```

16. Ecrire le code de la fonction isPositif qui renvoie 1 si a>0 et 0 sinon

```
int isPositif(double a);
```

17. Ecrire le code de la fonction moyenne qui calcule la moyenne de a,b,c

```
double moyenne(int a, int b, int c);
```

18. Ecrire le code de la fonction qui renvoie la valeur minimale entre a,b et c

```
int mini(int a, int b, int c);
```

19. Ecrire le code de la fonction isEspace qui renvoie 1 si c=' ' (caractère espace)

```
int isEspace(char c){
```

Les pointeurs

20. Modifier la valeur de a à 10 **en utilisant le pointeur**.

21. Déclarer une variable int a = 3; Afficher sa valeur et son adresse

22. Afficher la valeur de a **en utilisant le pointeur**.

```
int a = 5;
int *p;
```

23. Corriger le programme

```
int *p = &x;
int x = 10;
p = p + 3;
```

24. Quel est le résultat du programme

```
int x = 10;
int *p = &x;
*p = *p + 3;
printf("%d", x);
```

25. Quel est le résultat du programme

```
int a = 3;
int b = 6;
int *p = &a;
*p = 6;
p = &b;
*p = 9;
printf("%d %d", a, b);
```

26. Quel est le résultat du programme

```
int a = 3;
int b = 6;
int *p;
p = &b;
*p = 9;
printf("%d %d", *p, b);
```

27. Quel est le résultat du programme

```
int a = 3;
int *p = &a;
*p = *p * 2;
printf("%d %d", *p, a);
```

Les fonctions et pointeurs

28. Que fait le programme ci-dessous

```
void f(int *x) {
    *x = 10;
}
main(){
    int x=2;
    f(&x);
    printf("%d",x);
}
```

29. Que fait le programme ci-dessous

```
void f(int *x) {
    *x = *x + 10;
}
main(){
    int x=2;
    f(&x);
    printf("%d",x);
}
```

30. Que fait le programme ci-dessous

```
void f(int *y, int x) {
    *y = x + 10;
}
main(){
    int x=2;
    f(&x,2);
    printf("%d",x);
}
```

31. Que fait le programme ci-dessous

```
int f(int *y, int x) {
    *y = 3*x;
    return x*2;
}
main(){
    int y=2;
    y=f(&x,2);
    printf("%d %d",y,x);
}
```

32. Que fait le programme ci-dessous

```
int f(int *x) {
    *x = 3*x;
    return 2;
}
main(){
    int y=20;
    y=f(&x);
    printf("%d %d",y,x);
}
```

33. Que fait le programme ci-dessous

```
int f(int *x) {
    int y= *x;
    *x = 3*y;
    return y+2;
}
main(){
    int y=20;
    y=f(&x);
    printf("%d %d",y,x);
}
```

34. Que fait le programme ci-dessous

```
void f(int *x) {
    *x = 3* *x;
}
main(){
    int y=20;
    f(&x);
    printf("%d",x);
}
```

35. Que fait le programme ci-dessous

```
int f(int x) {
    int y= x*x;
    return y+2;
}
main(){
    int y=20;
    y=f(10);
    printf("%d",y);
}
```

36. Corriger le code ci-dessous :

```
void f(int *x) {
    *x = 3* *x;
}
main(){
    int x=20;
    f(x);
    printf("%d",x);
}
```

37. Corriger le code ci-dessous :

```
void f(int &x) {
    *x = 3* *x;
}
main(){
    int x=20;
    f(*x);
    printf("%d",x);
}
```

38. Ecrire le code de la fonction carre qui calcule le carre de x

```
void carre(int *x);
```

39. Ecrire l'appel de la fonction carre pour les valeurs de 1 à 9 (boucle for).

40. Ecrire le code de la fonction additionMoyenne qui calcule la moyenne des 2 valeurs (dans *moy) et renvoie le résultat de l'addition (return).

```
int additionMoyenne(double * moy, int a, int b);
```

41. Ecrire le code de la fonction additionMoyenne qui renvoie la moyenne des 2 valeurs (return) et calcule la somme dans *add.

```
double additionMoyenne(int * add, int a, int b);
```

42. Ecrire le code de doubleValeur qui ne double la valeur de *x.

```
void doubleValeur(int *x);
```

43. Corriger le code ci-dessous sachant que l'on veut calculer 3*x dans la fonction. Proposer une solution avec return

```
void f(int x) {
    x = 3* x;
}
main(){
    int x=20;
    f(x);
    printf("%d",x);
}
```

44. Corriger le code ci-dessous sachant que l'on veut calculer 3*x dans la fonction. Proposer une solution avec pointeur

```
void f(int x) {
    x = 3* x;
}
main(){
    int x=20;
    f(x);
    printf("%d",x);
}
```

45. Ecrire le code la fonction division qui prend 2 valeurs entières et doit renvoyer le quotient et le reste de la division, puis écrire l'appel de la fonction.
46. Ecrire le code de la fonction resolution_eq qui prend 3 paramètres a,b,c ($ax^2+bx+c=0$) d'une équation du second degré et doit renvoyer x1,x2 et un code permettant de préciser si l'équation possède des solutions ou pas.

Les tableaux et chaines de caractères

47. Créer un tableau d'entier de 10 éléments
48. Créer un tableau de double de 3 éléments et les initialiser avec les valeurs 1,2.2,3
49. Créer un tableau de caractères et les initialiser avec « bon »
50. Afficher un tableau d'entier de taille 10 (boucle for)
51. Compter le nombre d'éléments positifs dans un tableau d'entier
52. Compter le nombre de caractères de type chiffre dans la chaine de caractères
53. Compter le nombre de caractères d'une chaine de caractères
54. Saisir les valeurs d'un tableau d'entier (scanf et for)
55. Calculer la somme des éléments d'un tableau.
56. Recherche le minimum d'un tableau d'entier.

Les tableaux et pointeurs

57. Cet extrait de code est-il juste ? si oui quel est le résultat

```
int tab[3] = {1, 3, 4};
int *t=tab;
printf("%d %d",*t,*(t+2));
```

58. Cet extrait de code est-il juste ? si oui quel est le résultat

```
int *t= malloc(3,sizeof(int));
t[0]=1;
printf("%d",*t);
```

59. Cet extrait de code est-il juste ? si oui quel est le résultat

```
int tab[3] = {1, 3, 4};
tab[3]=1;
```

60. Cet extrait de code est-il juste ? si oui quel est le résultat

```
int tab[3] = {1, 3, 4}, somme=0;
for(int i=1;i<=3;i++){
    somme = somme + tab[i];
    printf("%d",tab[i]);
}
```

61. Cet extrait de code est-il juste ? si oui quel est le résultat

```
int *t= malloc(3,sizeof(int));
for(int i=0;i<3;i++)
    t[i]=0;
for(int i=0;i<3;i++)
    printf("%d",t[i]);
```

62. Cet extrait de code est-il juste ? si oui quel est le résultat

```
int tab[3] = {1, 3, 4};
for(int i=0;i<3;i++){
    tab[i]=0;
    printf("%d",tab[i]);
}
```

63. Cet extrait de code est-il juste ? si oui quel est le résultat

```
int tab[3] = {1, 3, 4};
for(int i=0;i<3;i++)
    *(tab+i)=0;
for(int i=0;i<3;i++)
    printf("%d",*(tab+i));
```

64. Cet extrait de code est-il juste ? si oui quel est le résultat

```
int *tab;
int n;
scanf("%d",&n) ;
tab = malloc(n *sizeof(int));
tab[0]=1;
printf("%d",*tab);
```

65. Cet extrait de code est-il juste ? si oui quel est le résultat

```
int tab[3];
int n;
scanf("%d",&n) ;
int tab[n];
tab[0]=1;
printf("%d",*tab);
```

66. Déclarer un pointeur qui pointe sur le début du tableau.

67. Afficher le premier élément avec le pointeur.

68. Afficher les 5 éléments du tableau **uniquement avec un pointeur** (pas de tab[i]).

69. Créer un tableau statique de 3 entiers et afficher les 3 valeurs (boucle for) en utilisant le formalisme pointeur

70. Créer un tableau dynamique de 3 entiers et afficher les 3 valeurs (boucle for) en utilisant le formalisme pointeur

71. Que fait le code ci-dessous ?

```
double *tab = malloc(10 * sizeof(double));
```

72. Cet extrait de code est-il juste ? si oui quel est le résultat

```
int tab[3] = {1,2,3};
int *p = tab;
printf("%d\n", *(p+1));
```

Les fonctions et tableaux

73. Appeler la fonction Somme1 dans le main, on crée un tableau de 4 éléments dans le main.

```
int Somme1(int tab[], int nb) {
    int somme = 0;
    tab[2] = 10;
    for (int i = 0; i < nb; i++)
        somme += tab[i]; //+tab[i]
    return somme;
}
```

74. Appeler la fonction Somme2 dans le main, on crée un tableau de 4 éléments dans le main.

```
int Somme2(int *tab, int nb) {
    int somme = 0;
    tab[2] = 10;
    for (int i = 0; i < nb; i++)
        somme += *(tab+i); //+tab[i]
    return somme;
}
```

75. Ecrire le code de la fonction echange et l'appel de la fonction. Quel sera le résultat affiché ?

```
void echange(int *a, int *b) {
    // à compléter
}
main(){
    int x=2,y=3;
    // appel de echange

    printf("%d %d",x,y);
}
```

76. Ecrire le code de la fonction incrementer

```
void incrementer(int *tab, int taille);
```

77. Ecrire le code de la fonction copier de dest vers src

```
void copier(int *dest, int *src);
```

78. Ecrire le code de la fonction afficherTableau qui affiche tous les éléments du tableau tab

```
void afficherTableau(int *tab, int taille);
```

79. Ecrire le code de la somme

```
int somme(int *tab, int taille);
```

80. Ecrire le code de la fonction calculMoyenne qui calcule la moyenne de tous les éléments du tableau.

```
double calculMoyenne(int tab[],int taille);
```

81. Ecrire le code de la fonction longueur qui renvoie la longueur de la chaîne de caractères

```
int longueur(char chaine[]);
```

82. Ecrire le code de la fonction nbLettres qui renvoie le nombre d'occurrence de lettre dans la chaîne

```
int nbLettres(char chaine[], char lettre);
```

83. Ecrire le code de minMaj qui copie tous les caractères de src dans dest mais en remplaçant toutes les lettres minuscules en majuscules

```
int minMaj(char *dest, char *scr);
```

84. Ecrire le code de test d'au moins 3 des fonctions données ci-dessus.
85. Ecrire une fonction qui compte combien de valeurs du tableau de taille n sont positives. Ecrire le code de la fonction et le programme de test (main).
86. Ecrire une fonction qui remplace tous les nombres négatifs par 0. Ecrire le code de la fonction et le programme de test
87. Ecrire une fonction qui multiplie tous les éléments par 2 (taille du tableau donnée en paramètre). Ecrire le code du main.
88. Ecrire un programme qui
- ✓ Demander à l'utilisateur un nombre n.
 - ✓ Créer un tableau dynamique de taille n.
 - ✓ appelle une fonction rempliTab qui vient remplir ce tableau avec des valeurs paires commençant par 100
 - ✓ appelle une fonction afficheTab qui affiche les éléments du tableau.
89. Ecrire le code des 2 fonctions rempliTab et afficheTab.
90. Ecrire le code de la fonction miroir(char* dst, char *src) qui prend la chaîne src et la recopie dans dst mais à l'envers.
91. Ecrire le code de la fonction maj(char* dst, char *src) qui prend la chaîne src et la recopie dans dst mais en majuscules (attention il faudra tester chaque caractère de la chaîne avant de le mettre en majuscule puisque 0 & W ne doivent pas être mis en majuscule)