

Nom : \_\_\_\_\_

Prénom : \_\_\_\_\_

Note : \_\_\_\_\_

A une question correspond une seule réponse juste, donc cocher la bonne réponse !

**Barème** : une bonne réponse = +1 - pas de réponse = 0 - une mauvaise réponse = -1

Q1 . Lequel de ces types de données permet de stocker le nombre **76.8** ?

char                       double                       long                       unsigned int

Q2 . Lequel de ces types de données peut stocker le nombre **-1000** ?

char                       unsigned double       long                       unsigned int

Q3 . Pour compiler le programme "**toto.c**", je peux utiliser :

`gcc -o toto toto.c`     `gcc toto`  
 `gcc -o toto.c toto`     `make toto.c`

Q4 . Si a et b sont définis par `int a = 0x45;` et `int b = 0x00;`, alors `(a || b)` vaut :

0                                       0x45                                       1

Q5 . Si a et b sont définis par `int a = 0x45;` et `int b = 0x00;`, alors `(a && b)` vaut :

0                                       0x45                                       1

Q6 . Combien vaut la variable "**nombre**" à la fin de ces opérations ?

```
int nombre;  
nombre = 10 % 3;
```

0                       10                       3                       3,333                       1                       ne se compile pas

Q7 . Qu'affiche ?

```
int a = 0;  
if(a != 0)  
    printf("a n'est pas nul !\n");
```

`a n'est pas nul !`                                       rien                                       ne se compile pas

Q8 . Qu'affiche l'instruction suivante : `printf("j'aime le C %d fois", 30/20 );`

- j'aime le C 0 fois                       j'aime le C 1 fois                       j'aime le C 1.5 fois  
 j'aime le C 2 fois                       je n'aime pas le C

Q9 . Qu'affichent les instructions suivantes :

```
int x = 9; int y = x + 10;  
printf("%d:%d:%d", x, y, y );
```

- 9:19:                       9:19:19                       9:19:%d                       %d:%d:%d                       x:y:y

Q10 . Qu'affiche ?

```
i = 2; j = 3;  
printf("%d %d", i++, ++j);
```

- 2 3                       3 4                       2 4                       3 3

Q11 . Qu'affiche ?

```
i = 5; printf("%s", i);
```

- 5                       erreur à la compilation                       erreur à l'exécution

Q12 . Que vaut **a** après l'exécution de ces lignes :

```
int a = 4;  
int b = a - 1;  
if(a%2)  
    a = b;  
else a = a / 2;
```

- 2                       3                       4

Q13 . Qu'affiche ?

```
int variable = 1;  
switch (variable)  
{  
    case 1: printf("1"); break;  
    case 2: printf("2"); break;  
}
```

- 1                       2                       12

Q14 . Qu'affichera ce code ?

```
int majeur = 0;
long argentEnPoche = 10000, age = 19;

majeur = !(age < 18);

if (argentEnPoche > 10000 && majeur)
    printf("Vous pouvez ouvrir un compte en banque !");
else
    printf("Sortez d'ici ou j'appelle la securite");
```

- rien  Vous pouvez ouvrir un compte en banque !  
 Sortez d'ici ou j'appelle la securite  ne se compile pas

Q15 . Qu'affiche ?

```
int variable = 12;
switch (variable)
{
    case 5:
        printf("Salut");
    case 12:
        printf("Bonjour");
    default:
        printf("Bonsoir");
}
```

- Salut  BonjourBonsoir  
 Bonjour  erreur à la compilation

Q16 . 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 = 0 ; compteur < 9 ; compteur += 2)  
 for (compteur = 1 ; compteur < 9 ; compteur += 2)  
 for (compteur = 1 ; compteur <= 7 ; compteur+2)  
 for (compteur = 1 ; compteur < 8 ; compteur++)

Q17 . Combien de fois le message "**Salut**" sera-t-il affiché ici ?

```
long compteur = 14;

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

- 0 fois                                       14 fois                                       1 fois  
 15 fois                                       C'est une boucle infinie

Q18 . Qu'affichent les instructions suivantes :

```
int i = 10;
while (i > 0)
{
    i = i - 4;
    printf("%d ", i);
}
```

- 10 6 2                                       6 2 -2                                       6 2

Q19 . Quelle est la valeur de **i** après exécution des instructions suivantes :

```
int i;
int s = 0;
for (i=1; i < 6; i=i+2) s = s + i;
```

- 4                       6                       8                       7                       5

Q20 . Combien de fois le message "**Salut**" sera-t-il affiché ici ?

```
long compteur = 14;

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

- 0 fois                                       14 fois                                       1 fois  
 15 fois                                       C'est une boucle infinie