

## Projet-TP [S1]

Nom : ..... Prénom : ..... Groupe : .....

---

Q1) Ecrivez un programme qui lit trois valeurs entières (A, B et C) au clavier et qui affiche la plus grande des trois valeurs, en utilisant **uniquement des opérateurs conditionnels** sans variable d'aide.

e.g. (B>C)? B:C ; cela signifie si B > C renvoie A sinon renvoie B.

Q2) Que ferait l'expression ci-dessous ?

```
c = getchar() != EOF ;
```

Q3) Pourquoi la variable utilisée pour contenir la valeur de retour de **getchar** doit-elle être de type int ?

Q4) Quelle est la différence entre les formes préfixe et suffixe de l'opérateur ++?

Q5) Que ferait l'expression ci-dessous ?

```
i = i ++ ;
```

Q6) Quelle est la définition d'une chaîne de caractères en C ?

Q7) Soit l'instruction :

```
#define N 10;
```

Y'a-t-il un problème avec celle-là ? justifier avec exemple ?

Q8) Supposons que vous ayez défini la macro

```
#define SIX 2 * 3
```

Ensuite, supposons que vous l'avez utilisé dans une autre expression :

```
int x = 12 / SIX;
```

Quelle serait la valeur de x ? justifier ?

Q9) Soit les deux lignes :

```
int i = 10;
```

```
#define J 10
```

Quelles sont les différences entre i et J ?

Q10) Quels sont les deux types de division différents que l'opérateur / peut effectuer ? Dans quelles circonstances effectue-t-il chacun ?

Q11) Quelles sont les définitions des valeurs `` booléennes " true et false en C ?

Q12) Nommez trois utilisations du point-virgule en C.

Q13) Soit la boucle suivante:

```
for(i = 0; i < 10; i = i + 1)  
    printf("i is %d\n", i);
```

Donnez le code équivalent en utilisant une boucle while ?

Q14) Quel résultat affiche le code suivant ?

```
int x = 3;  
if(x)  
    printf("yes\n");  
else    printf("no\n");
```

Q15) Quel résultat affiche le code suivant ?

```
int i;
for(i = 0; i < 3; i = i + 1)
    printf("a\n");
    printf("b\n");

printf("c\n");
```

Q16) Quelles sont les utilisations des commentaires ?

Q17) Quelles sont les valeurs les plus grandes et les plus petites qui peuvent être stockées de manière fiable dans une variable de type int ?

Q18) Pourquoi l'indentation (indentation) est importante ? Avec quelle attention le compilateur y prête-t-il attention ?

Note :

L'indentation consiste à introduire une ou plusieurs tabulations au début de certaines lignes d'un programme afin d'en améliorer la lisibilité.

Voici un exemple :

```
While(i<100){
If(a%2 =0){
Printf(«a est apire »);
compteurPaire++;
}else{
Printf(«a est impair »);
compteurImpaire++;
}
}
```

```
While(i<100){
    If(a%2 =0){
        Printf(«a est apire »);
        compteurPaire++;
    }else{
        Printf(«a est impair »);
        compteurImpaire++;
    }
}
```

Q19) Qu'impriment ces boucles ?

a. for(i = 0; i < 10; i = i + 2)  
printf("%d\n", i);

b. for(i = 100; i >= 0; i = i - 7)  
printf("%d\n", i);

c. for(i = 1; i <= 10; i = i + 1)  
printf("%d\n", i);

d. for(i = 2; i < 100; i = i \* 2)  
printf("%d\n", i);

Q20) Écrivez un programme pour imprimer ce triangle :

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

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

**N'utilisez pas dix instructions printf;** utilisez plutôt **deux boucles imbriquées**. Vous devrez utiliser des accolades autour du corps de la boucle externe si elle contient plusieurs instructions

:

```
pour (i = 1; i <= 10; i = i + 1)
{
    /* plusieurs déclarations */
    /* peut entrer ici */
}
```