

Question I (4 points)

- 1) Expliquer brièvement les notions de conversions implicites et conversions systématiques ?
- 2) Quelle est la différence entre une fonction récursive directe et croisée ?
- 3) Quel est la différence entre les boucles *while* et *do while* ?
- 4) Comment peut-on réaliser une boucle infinie avec *for* ?

Question II (4 points)

<pre>int i=7; float f=5.5; char c='w'; (119) //Evaluer les expressions suivantes : (1) i + f <= 1 (2) c >= 10 * (i + f) (3) i >= 6 && (c == 'w') (4) i >= 6 (c == 119) (5) f < 11 && (i > 100) //Evaluer les expressions suivantes : a = 16; b = 2; c = 10; (6) c += a > 0 && a <= 15 ? ++a : a/b; (7) c = a > 30 ? b = 11 : (c = 100);</pre>	<pre>#include <stdio.h> main () { int n = 123 ; int p = 5 ; float x = 356.41; printf("B : %d %2f\n", n, x) ; printf("C : %d %3f\n", n, x) ; printf("D : %10.3f %10.3e\n", x, x) ; printf("E : %-5d %f\n", n, x) ; printf("F : %*d\n", p, n) ; printf("G : %*.f\n", 9, 3, x) ; }</pre>
---	---

Question III (5 points)

Soit un tableau de réels *T* de dimension *dim*.

Écrire une fonction *récursive* qui prend comme argument un réel *val* et la dimension *dim* et renvoie la position de *val* dans le tableau *T* s'il existe et -1 sinon.

Question IV (4 points)

Ecrire un programme qui pour une valeur de *n* choisie au clavier, affiche *n* fois la valeur *n* sur la *n*-ème ligne. Pour *n* = 5, on a par exemple l'affichage suivant :

```
1
22
333
4444
55555
```

Question V (3 points)

Ecrire une fonction qui renvoie la somme des inverses des éléments d'un tableau d'entiers, utilisez les boucles.

CC 2018 : Langage C

(1)

Question 1 :

1 - conversion systématique : se fait lorsqu'il y a des opérandes de ~~deux~~ type short ou char, les opérateurs arithmétiques dans C ne se font jamais pour ces deux types d'abord, ils sont convertis au type int,

conversion implicite : les opérations en C, se font pour deux opérandes de ~~de~~ même type, alors que pour une opération de deux opérandes de type différent, l'un de les deux est converti au type du deuxième à condition de ne pas perdre la précision, en général se fait sous l'ordre :
int \Rightarrow long \Rightarrow float \Rightarrow double \Rightarrow long double

2 - fonction récursive directe : dans sa définition elle fait appelle à elle même.

fonction récursive croisée : dans sa définition elle fait appelle à une autre fct, qui fait l'appelle à la même fonction. (un cycle peut contenir plus de deux fcts).

3 - la boucle do-while s'exécute au moins une fois, car la condition est évaluée après le bloc d'instruction, par contre while, si la condition ~~est~~ n'est pas vérifiée, il n'exécute aucune fois

4 - sans condition $\{ \dots \}$

Question II :

(2)

1 - 0

4 - 1

2 - 0

5 - 0

3 - 1

6 - 1

7 - 100

// ^ espace.

B : 123 ^ 356.410000

C : 123 356.410000

D : ^ ^ 356.410 3.564e+002

E : 123 ^ 356.410000

F : ^ 123

G : ^ 356.410

Question III :

// T déjà déclaré de dimension dim (global),
int chercher (float val, int dim) {

if (dim >= 1) {

if (val == T[dim-1]) {

return (dim-1);

} else { return (chercher(val, dim-1));

} else { return (-1); }

}

Question IV :

③

```

main() {
    int i, j, n;
    printf("saisir la valeur de n\n");
    scanf("%d", &n);
    for(i=1; i<=n; i++) {
        for(j=1; j<=i; j++) {
            printf("%d", i);
        }
        printf("\n");
    }
}

```

Question V :

la somme des inverses des éléments d'un tableau.

```

float som_inv(int T[100]) {
    int i;
    float s;
    for(i=0; i<100; i++) {
        s = s + (float)1/T[i]; // suppose T[i] != 0
    }
    return(s);
}

```