

## Algorithmique et Structures de données Corrigé de l'Examen du 17 Janvier 2022

### Q1 : Fonction récursive pour calculer le PGCD

(4 pts)

```
void pgcd (int i, int j)
{
    int k ;
    k = i % j /* reste de la division de i par j */
    if (k == 0) /* base : i est un multiple de j*/
        return j;
    else /* appel récursif*/
        return pgcd (j, k);
}
```

### Q2 : Tri par insertion

(6 pts)

(a) Illustration du tri par insertion à partir du tableau  $v = \{6, 4, 8, 3, 2, 5, 1, 7\}$

(2 pts)

<b>Input</b>	: 6 4 8 3 2 5 1 7
Itération 1	: 4 6 8 3 2 5 1 7
Itération 2	: 4 6 8 3 2 5 1 7
Itération 3	: 3 4 6 8 2 5 1 7
Itération 4	: 2 3 4 6 8 5 1 7
Itération 5	: 2 3 4 5 6 8 1 7
Itération 6	: 1 2 3 4 5 6 8 7
Itération 7	: 1 2 3 4 5 6 7 8
<b>Output</b>	: 1 2 3 4 5 6 7 8

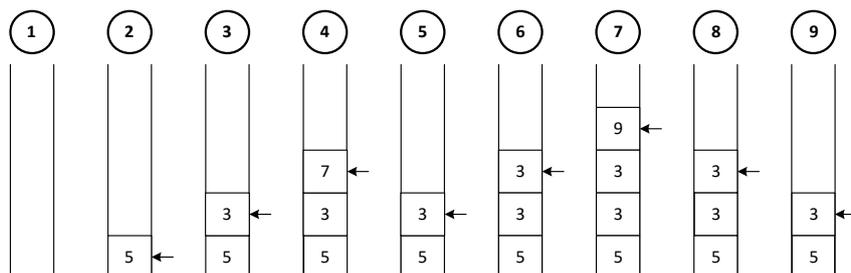
(b) Réalisation de l'algorithme de tri par insertion

(4 pts)

```
void tri_insertion (float t[], int n)
{
    int i, k ;
    float x ;
    for (i = 1; i < n; i++)
    {
        k = i - 1;
        x = t[i];
        while (k >= 0 && t[k] > x)
            t[k + 1] = t[k], k --;
        t[k + 1] = x;
    }
}
```

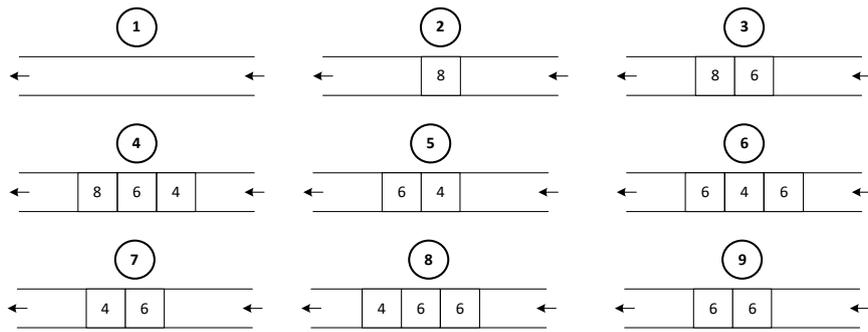
### Q3 : (a) Etat de la pile

(2 pts)



**Q3 : (a) Etat de la file**

(2 pts)



**Q4 : (a) Déclaration de types pour une liste d'entiers implémentée en utilisant une liste simplement chaînée (LSC).**

(2 pts)

```

struct Cellule
{
    int contenu ;
    struct Cellule *suivant ;
} ;
typedef struct Cellule Cellule, *Liste ;
    
```

**Q4 : (b) Fonction qui insère un nouvel élément après une certaine position dans une liste LSC.**

(4 pts)

```

Liste insert_position (Liste a, int x, int pos)
{
    int k = 1 ;
    Liste b ;

    b = (Liste) malloc (sizeof (Cellule));
    b->contenu = x;

    if (EstListeVide (a) || pos == 0) /*insertion au début*/
    {
        b->suivant = a;
        return b;
    }
    /* avancer jusqu'à la position pos ou au dernier élément */
    while (!EstListeVide (a->suivant) && k < pos)
    {
        a = a->suivant;
        k = k + 1;
    }

    if (!EstListeVide (a->suivant)) /* insertion après position pos*/
        b->suivant = a->suivant;

    else /* insertion à la fin */
        b->suivant = NULL ;

    a->suivant = b;

    return a ;
}
    
```