

COMBETTE
ElisePIL
2013

19,25

Mardi 20 MaiDE Programmation③ Exercice 1 Passage de paramètres

1. void échanger (float* p1, float* p2)

```
{  
    float tempo;  
    tempo = *p1;  
    *p1 = *p2;  
    *p2 = tempo;  
}
```

2. int main()

```
{  
    float A, B;  
    A = 10,5;  
    B = 19,75;  
    float* pA = &A;  
    float* pB = &B;  
    échanger (pA, pB);  
    return 0;  
}
```


2.25

Exercice 2 chaîne de caractères.

```
char s1[30];
char s2[30];
int ok = 1; // "vrai"
gets(s1);
gets(s2);
int i1 = 0;
int i2 = 0;
while ((s2[i2] != '\0') && (ok == 1))
{
    i2 = 0;
    while ((s1[i1] != '\0') && (ok == 1))
    {
        if (s1[i1] == s2[i2]) ok = 0; // "faux"
        i1++;
        ok = !ok;
        i2++;
    }
}
```

8 Exercice 3 Tableau

2.1. void initialiser (int * t, int taille)

```
{
    srand(time(NULL)); // générateur de nombres aléatoires. A ne faire qu'une fois
                        // → dans le main.
    int i;
    for (i = 0; i < taille; i++) t[i] = rand() % (10 * taille n + 1);
    int tempo, i1, i2;
```



```
for (i1=0; i1 < taille; i1++)
```

```
{
```

```
    for (i2=0; i2 < taille; i2++)
```

```
    {
```

```
        if (t[i1] > t[i2])
```

```
        {
```

```
            tempo = t[i1];
```

```
            t[i1] = t[i2];
```

```
            t[i2] = tempo;
```

```
        }
```

```
    }
```

```
}
```

2. 2. int estCroissant (int* t, int taille)

```
{
```

```
    int bool = 1;    // vrai
```

```
    int i
```

```
    while ((bool == 1) && (i < taille-1))
```

```
    {
```

```
        if (t[i] > t[i+1]) bool = 0;    // faux.
```

```
        i++;
```

```
    }
```

```
    return bool;
```

```
}
```

2. 3. int* clone (int* t, int taille)

```
{
```

```
    int* tclone;
```

```
    tclone = malloc (taille * sizeof(int));
```

```
    int i;
```

```
    for (i=0; i < taille; i++) tclone[i] = t[i];
```

```
    return tclone;
```

```
}
```



```

2 4. int main()
    {
        int t[10];
        initialiser(t, 10);
        int* this;
        this = clone(t, 10);
        int croissant;
        croissant = estCroissant(this, 10);
        if (croissant == 1) printf("Le tableau est croissant. \n");
        else printf("Le tableau n'est pas croissant. \n");
        for (croissant = 0; croissant < 10; croissant++) printf("%d \n", this[croissant]);
        // on réutilise la variable croissant pour ne pas en créer une nouvelle.
        return 0;
    }

```

③ Exercice 4 Matrice

```

2 1. int damier[10][10];
    damier[0][0] = 0;
    int l, c;
    for (l = 1; l < 10; l++) // on initialise la 1ère colonne ...
    {
        if (damier[l-1][0] == 0) damier[l][0] = 1;
        else damier[l][0] = 0;
    }
    for (l = 0; l < 10; l++) // ... puis on initialise le reste du damier.
    {
        for (c = 1; c < 10; c++)
        {
            if (damier[l][c-1] == 0) damier[l][c] = 1;
            else damier[l][c] = 0;
        }
    }

```



```

4.2. for(l=0; l<10; l++)
    {
        for(c=0; c<10; c++)
        {
            if(damier[l][c] == 1) printf("%c", 219);
            else printf(" ");
        }
        printf("\n");
    }

```

③ Exercice 5 Récursivité

1. cette somme correspond à la relation de récurrence :

$$\begin{cases} u_0 = 1 \\ u_{n+1} = u_n + \frac{1}{2(n+1)+1} \end{cases} \quad \forall n \in \mathbb{N}.$$

2. float somme (int n)

```

{
    if(n == 1) return 1.0; // on ne veut que le 1er terme donc d'indice 0.
    else return (1.0 / (2 * (n-1) + 1) + somme(n-1));
}

```

3. printf("%f", somme(n));





