

Exercices supplémentaires

L1 CUPGE 2013 - Université d'Aix-Marseille

Exercice 1 (Suite de Syracuse)

```
#include<stdio.h>

void syracuse(int n)
{
    while(n!=1)
    {
        if (n%2 == 0)
            n = n/2;

        else
            n = 3*n+1;

        printf("n vaut %d\n", n);
    }
    return;
}

int main(void)
{
    int n;
    printf("Entrez une valeur pour n :\n");
    scanf("%d",&n);
    syracuse(n);
    return 0;
}
```

Exercice 2 (Tri bulle)

```
void bulle(int t[], int n)
{
    int i, temp;
    for (i=0; i<n-1; i=i+1)
    {
        if (t[i] > t[i+1])
        {
            temp = t[i];
            t[i] = t[i+1];
            t[i+1] = temp;
        }
    }
    return;
}
```

Notez qu'on s'arrête à $i = n-2$ puisqu'on utilise la case $i+1$ et que la dernière case est la case n .

Exercice 3 (Décomposition en facteurs premiers)

```
void decompose(int k, int facteurs[])
{
    int i = 2, j = 0;
    while (k != 1)
    {
        if (k%i == 0)
        {
            facteurs[j] = i;
            j = j+1;
        }
        else
        {
            i = i+1
        }
    }
    return;
}
```

Exercice 4 (Lecture de code)

1; 3; 2; 4; 2; 5 affiche 1, 2, 4, 5;
4; 3; 2; 1; 2; 6 affiche 4, 6.

La fonction affiche seulement les nombres supérieurs à tous ceux qui précèdent, autrement dit, la plus grande sous-suite rangée par ordre croissant qui part du début.

Exercice 5 (Introduction à la bioinformatique)

```
int mystere (char tab[], int n, char mot[], int m)
{
    int i,j;
    for (i=0; i<n-m; i=i+1)
    {
        trouve = 1;
        for (j=0; j<m; j=j+1)
        {
            if (tab[i+j] != mot[j])
                trouve = 0;
        }

        if (trouve == 1)
            return 1;
    }
    return 0;
}
```

Exercice 6 (Échiquier)

```
#include<stdio.h>
#define N 243
#define M 548

int main(void)
{
    int i,j;
    int mat[N][M];
    for (i=0; i<N; i = i+1)
    {
        for (j=0; j<M; j=j+1)
        {
            if ((i+j)%2 == 0)
                mat[i][j] = 0;
            else
                mat[i][j] = 1;
            printf("%d ",mat[i][j]);
        }
        printf("\n");
    }

    return 0;
}
```

Entrée-sortie sur des fichiers

Exercice 7 (Personne n'entend l'espace crier)

```
#include<stdio.h>

int main(void)
{
    FILE *f;
    int nbmots = 0, car = 0;
    f = fopen("livre.txt", "r");
    while (car != -1)
    {
        car = getc(f);
        if (car == ' ')
            nbmots = nbmots + 1;
    }
    printf("Le nombre de mots est %d\n", nbmots);
    fclose(f);
    return 0;
}
```

Exercice 8 (Formation au métier de caissier)

```
#include<stdio.h>

int main(void)
{
    int client, argent, err, x;
    printf("Entrez le numero du client :\n");
    scanf("%d", &client);
    FILE *f;
    f = fopen("clients.txt", "r");
    while (err == 1)
    {
        err = fscanf(f, "%d", &x);
        if (x == client)
        {
            err = fscanf(f, "%lf", &argent);
            printf("Le client doit %lf euros.\n", argent);
            fclose(f);
            return 0;
        }
        else
            err = fscanf(f, "%lf", &argent);
    }
    fclose(f);
}
```

```

    printf("Le client n'a pas ete trouve.\n");
    return 0;
}

```

Plus difficile

Exercice 9 (Mots mêlés)

```

#include<stdio.h>

void demeler (char mat[50][50], char mot[], int n)
{
    int horizontal, vertical, diagonal
    for(i=0; i<50; i=i+1)
    {
        for(j=0; j<50; j=j+1)
        {
            horizontal = 1;
            if (i+n >= 50)
                horizontal = 0;
            else
            {
                for (k=0; k<n; k=k+1)
                {
                    if (mat[i+k][j] != mot[k])
                        horizontal = 0;
                }
            }
            if (horizontal == 1)
            {
                printf("Le mot a ete trouve horizontalement a la case (%d, %d).\n", i,j);
                return;
            }

            vertical = 1;
            if (j+n >= 50)
                vertical = 0;
            else
            {
                for (k=0; k<n; k=k+1)
                {
                    if (mat[i][j+k] != mot[k])
                        horizontal = 0;
                }
            }
            if (vertical == 1)

```

```

    {
        printf("Le mot a ete trouve vericalement a la case (%d, %d).\n", i,j);
        return;
    }

    diagonal = 1;
    if ((i+n >= 50) || (j+n >= 50))
        diagonal = 0;
    else
    {
        for (k=0; k<n; k=k+1)
        {
            if (mat[i+k][j+k] != mot[k])
                diagonal = 0;
        }
    }
    if (diagonal == 1)
    {
        printf("Le mot a ete trouve diagonalement a la case (%d, %d).\n", i,j);
        return;
    }
}

}

printf("Le mot n'a pas ete trouve.\n");
return;
}

```