

AC 13.03 – Traduire un algorithme, dans un langage et pour un environnement donné.

Ressources: SAE 1.05 - R1.07

Contexte : Dans le cadre de ma formation en BUT Réseaux et Télécommunications, j'ai réalisé une activité liée à la compétence AC13.03 – Traduire un algorithme, dans un langage et pour un environnement donné.

L'objectif était de pouvoir créer, en autonomie, un programme en C et/ou en Python.

Savoir mis en œuvre → Langage informatique (python et C). Installation de module VScode pour l'exécution des programmes.

Savoir-faire mis en œuvre → Création de programme (en Python et en C) en utilisant, au besoin, des boucles et des fonctions tout en manipulant des tableaux ou des fichiers "texte". Veillez à la bonne exécution de ce programme et au bon déroulement de son procédé (vérifier que le programme fasse ce que l'on veut). Savoir analyser et comprendre un algorithme.

Savoir-être mis en œuvre → La rigueur et l'autonomie pour la création et la recherche d'erreurs au sein du programme créé.

Tâche réalisée et les résultats → J'ai dû réaliser de nombreux programmes au cours de la formation. Voici quelques exemples de programmes que j'ai réalisés. À noter que l'ensemble des programmes suivants étaient à réaliser en C et en Python (excepté le projet).

1. Je devais réaliser un programme qui nous demandait de rentrer un jour, un mois et une année et qui nous renvoyait de quel jour de la semaine il s'agissait.

```
Entrez le jour : 12
Entrez le mois : 09
Entrez l'année : 2007
Le jour de la semaine est : Mercredi
```

Code que j'ai réalisé en Python:

```
def jour(jours, mois, annee):
    valeurs_mois = [1, 4, 4, 0, 2, 5, 0, 3, 6, 1, 4, 6]
    jours_semaine = ["Samedi", "Dimanche", "Lundi", "Mardi", "Mercredi", "Jeudi", "Vendredi"]

    deux_derniers = annee % 100
    valeur = deux_derniers + (deux_derniers // 4) + valeurs_mois[mois - 1]

    bissextile = (annee % 4 == 0 and annee % 100 != 0) or (annee % 400 == 0)
    if bissextile and mois in [1, 2]:
        valeur -= 1

    if 1600 <= annee < 1700:
        valeur += 6
    elif 1700 <= annee < 1800:
        valeur += 4
    elif 1800 <= annee < 1900:
        valeur += 2
    elif 1900 <= annee < 2000:
        valeur += 0
    elif 2000 <= annee < 2100:
        valeur += 6
    elif 2100 <= annee < 2200:
        valeur += 4

    valeur += jours % 7
    return jours_semaine[valeur % 7]

jours = int(input("Entrez le jour : "))
mois = int(input("Entrez le mois : "))
annee = int(input("Entrez l'année : "))

resultat = jour(jours, mois, annee)
print("Le jour de la semaine est :", resultat)
```

Code que j'ai réalisé en C:

```
const char* jour(int jours, int mois, int annee){
    int valeurs_mois[] = {1, 4, 4, 0, 2, 5, 0, 3, 6, 1, 4, 6};
    char* jours_semaine[] = {"Samedi", "Dimanche", "Lundi", "Mardi", "Mercredi", "Jeudi", "Vendredi"};

    int deux_derniers = annee % 100;
    int valeur = deux_derniers + (deux_derniers/4) + valeurs_mois[mois - 1];

    int bissextile = (annee % 4 == 0 && annee % 100 != 0) || (annee % 400 == 0);
    if (bissextile && (mois == 1 || mois == 2)){
        valeur -=1;
    }
    if (annee >= 1600 && annee < 1700){
        valeur += 6;
    }
    else if (annee >= 1700 && annee < 1800){
        valeur += 4;
    }
    else if (annee >= 1800 && annee < 1900){
        valeur += 2;
    }
    else if (annee >= 1900 && annee < 2000){
        valeur += 0;
    }
    else if (annee >= 2000 && annee < 2100){
        valeur += 6;
    }

    else if (annee >= 2100 && annee < 2200){
        valeur += 4;
    }
    valeur += jours % 7;
    return jours_semaine[valeur % 7];
}

void main(){
    int jours, mois, annee;

    printf("Entrez le jour : ");
    scanf("%d", &jours);

    printf("Entrez le mois : ");
    scanf("%d", &mois);

    printf("Entrez l'année : ");
    scanf("%d", &annee);

    const char* jourSemaine = jour(jours, mois, annee);
    printf("Le jour de la semaine est : %s\n", jourSemaine);
    return 0;
}
```

2. Je devais modéliser les **effets de la gravité** sur un objet qui tombe depuis une tour de hauteur donnée en mètres. Je voulais également que le programme m'affiche la hauteur à laquelle se trouvait l'objet tous les x mètres pendant sa chute.

```
Simulation de la chute :
temps = 0.0 s : hauteur = 100.0 m
temps = 1.0 s : hauteur = 95.096675 m
temps = 2.0 s : hauteur = 80.3867 m
temps = 3.0 s : hauteur = 55.870075 m
temps = 4.0 s : hauteur = 21.546800000000005 m
temps = 5.0 s : hauteur = 0.00 m → Boum!!!
```

Code que j'ai réalisé en Python:

```
def chute(hauteur):
    g = 9.80665
    intervalle = float(input("Entrez l'intervalle de temps (en secondes) : "))

    t = 0.0
    print("\nSimulation de la chute :")
    while True:
        distance = 0.5 * g * t**2
        hauteur_restante = hauteur - distance

        if hauteur_restante <= 0:
            print("temps = ", t, "s : hauteur = 0.00 m → Boum!!!")
            break
        else:
            print("temps = ", t, "s : hauteur = ", hauteur_restante, "m")

        t += intervalle

chute(100)
```

Code que j'ai réalisé en C:

```
#include <stdio.h>

void chute(double hauteur)
{
    double g = 9.80665;
    double intervalle;
    double t = 0.0;

    printf("Entrez l'intervalle de temps (en secondes) : ");
    scanf("%lf", &intervalle);

    while (intervalle <= 0){
        double distance = 0.5 * g * t * t;
        double hauteur_restante = hauteur - distance;

        if (hauteur_restante <= 0) {
            printf("temps = %f s : hauteur = 0.00 m → Boum!!!\n", t);
            break;
        } else {
            printf("temps = %f s : hauteur = %f m\n", t, hauteur_restante);
        }

        t += intervalle;
    }

    int main() {
        chute(100.0);
        return 0;
    }
}
```

3. Je devais également recréer le **jeu de l'allumette** (Chaque joueur prend à tour de rôle de 1, 2 ou 3 allumettes dans un tas comprenant initialement x allumettes. Celui qui prend la dernière allumette perd le jeu). Je devais mettre en place que l'un des deux joueurs soit l'ordinateur.

Code que j'ai réalisé en Python:

```
import random

def allumette2(nb_allumette, joueur_depart):
    joueur = joueur_depart
    joueur_humain = int(input("Quel joueur voulez-vous être ? (1 ou 2) : "))
    joueur_ordi = 2 if joueur_humain == 1 else 1

    while nb_allumette > 0:
        print(f"\nLe nombre restant d'allumettes dans le jeu est égal à {nb_allumette}")

        if joueur == joueur_humain:
            prise = int(input("Quel est le nombre d'allumettes que vous voulez retirer ? "))
        else:
            prise = random.randint(1, min(3, nb_allumette))
            print(f"Le nombre d'allumettes prises par l'ordinateur est égal à {prise}")

        if prise < 1 or prise > 3 or prise > nb_allumette:
            print("\nVotre choix n'est pas valide.")

        nb_allumette -= prise

        if nb_allumette == 0:
            if joueur == joueur_humain:
                print("\nL'ordinateur a gagné.")
                print("Vous avez perdu.")
            else:
                print("\nBravo ! Vous avez gagné.")
                print("L'ordinateur a perdu.")
            break

    joueur = 2 if joueur == 1 else 1

x = int(input("Nombre d'allumettes au départ : "))
joueur_initial = int(input("Quel joueur commence ? (1 ou 2) : "))
allumette2(x, joueur_initial)
```

Code que j'ai réalisé en C:

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

void allumette2(int nb_allumette, int joueur_depart) {
    int joueur = joueur_depart;
    int joueur_humain, joueur_ordi, prise;
    int fin_du_jeu = 0;

    printf("Quel joueur voulez-vous être ? (1 ou 2) : ");
    scanf("%d", &joueur_humain);

    if (joueur_humain == 1) {
        joueur_ordi = 2;
    } else {
        joueur_ordi = 1;
    }

    while (!fin_du_jeu) {
        printf("Le nombre restant d'allumettes dans le jeu est égal à %d\n", nb_allumette);

        if (joueur == joueur_humain) {
            printf("Quel est le nombre d'allumettes que vous voulez retirer ?");
            scanf("%d", &prise);
        } else {
            prise = 1 + rand() % ((nb_allumette < 3) ? nb_allumette : 3);
            printf("Le nombre d'allumettes prises par l'ordinateur est égal à %d\n", prise);
        }

        if (prise < 1 || prise > 3 || prise > nb_allumette) {
            printf("Votre choix n'est pas valide.\n");
        } else {
            nb_allumette -= prise;

            if (nb_allumette == 0) {
                if (joueur == joueur_humain) {
                    printf("L'ordinateur a gagné.\n");
                    printf("Vous avez perdu.\n");
                } else {
                    printf("Bravo ! Vous avez gagné.\n");
                    printf("L'ordinateur a perdu.\n");
                }
                fin_du_jeu = 1;
            } else {
                if (joueur == 1) {
                    joueur = 2;
                } else {
                    joueur = 1;
                }
            }
        }
    }
}

int main() {
    int x, joueur_initial;
    srand(time(NULL));
    printf("Nombre d'allumettes au départ : ");
    scanf("%d", &x);
    printf("Quel joueur commence ? (1 ou 2) : ");
    scanf("%d", &joueur_initial);

    allumette2(x, joueur_initial);
    return 0;
}
```

Partie projet. Dans ce projet, je devais réaliser un programme en C puis un autre en Python. Les deux sont totalement distincts.

- Dans le premier programme en C, je devais trier un tableau de trois manières différentes, puis comparer le temps que les fonctions mettaient à le trier. Je disposais de certains bouts de code pour me guider. Je devais donc compléter le code existant puis le continuer pour qu'il réalise la tâche demandée.

```
Tri par selection : 12.723000 secondes
Tri a bulles : 33.055000 secondes
Tri a peigne : 0.018000 secondes
```

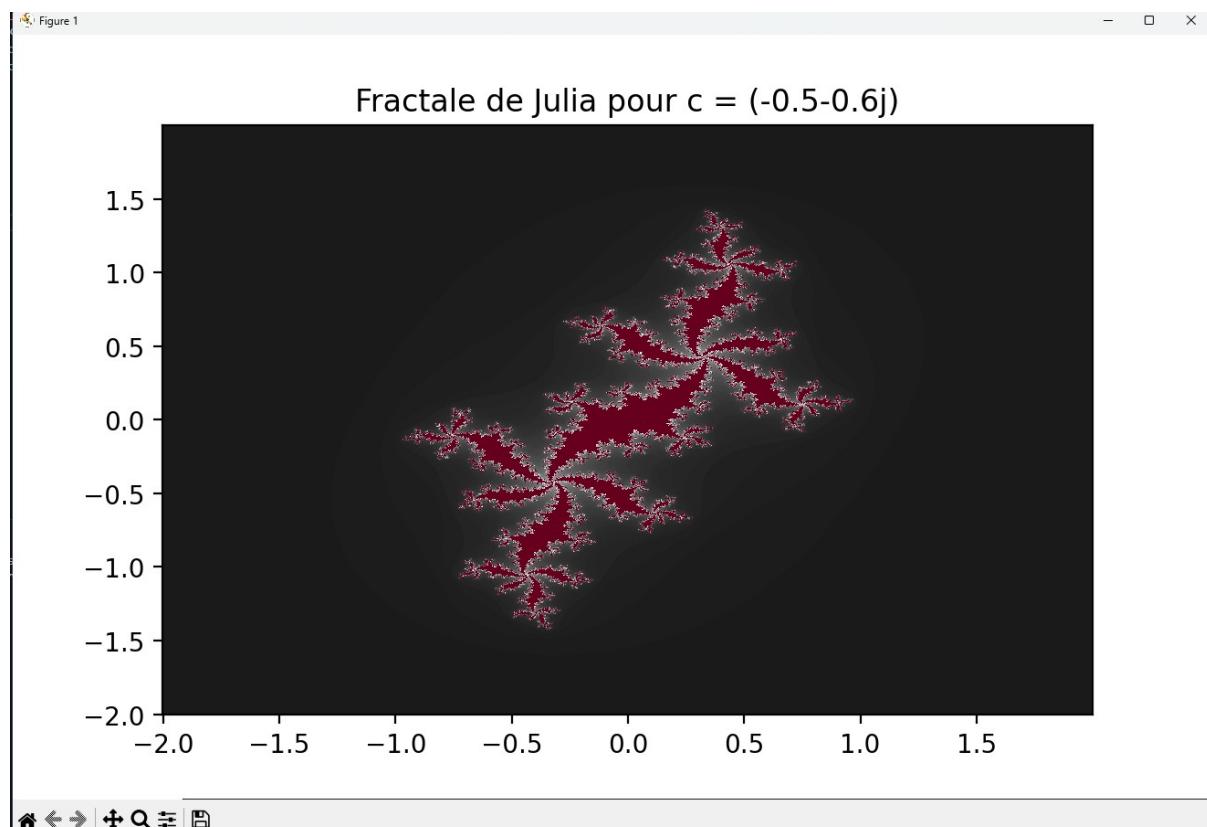
Code des trois tri présent:

```
void tri_selection(int T[], int n){
    for (int i = 0; i < n - 1; i++){
        int indice_min = i;
        for (int j = i + 1; j < n; j++){
            if (T[j] < T[indice_min]){
                indice_min = j;
            }
        }
        echanger(T+i, T+indice_min);
    }
}

void tri_bulles(int T[], int n){
    bool ech = true;
    while (n > 0 && ech==true){
        ech = false;
        int j = 0;
        while (j < n-1){
            if (T[j] > T[j + 1]){
                echanger(&T[j], &T[j + 1]);
                ech = true;
            }
            j++;
        }
        n--;
    }
}

void tri_peigne(int T[], int n){
    bool ech = true;
    int h = n;
    while (h > 1 || ech == true){
        h = (int)(h / 1.3);
        if (h < 1) {
            h=1;
        }
        ech = false;
        int j=0;
        while (j<n-h){
            if (T[j] > T[j + h]){
                echanger(&T[j], &T[j + h]);
                ech = true;
            }
            j++;
        }
    }
}
```

- Dans mon deuxième programme en Python, je devais tracer une fractale. Le début des codes était donné et je devais les compléter.



Les problèmes que j'ai rencontrés → Je n'ai pas rencontré de réel problème dans cette ressource. J'ai tout de même dû m'aider d'internet à certains moments pour la réalisation de mes programmes en C.

Point faible / Point à améliorer → Encore beaucoup d'erreurs et de recherches réalisées pendant la réalisation pour les programmes en C