Exercice 1 : Lanceur Spatial

0.1 Contexte

L'équation de Constantin Tsiolkovski est l'équation fondamentale de l'astronautique reliant l'accroissement de vitesse au cours d'une phase de propulsion d'un astronef doté d'un moteur à réaction au rapport de sa masse initiale à sa masse finale.

Elle permet de rapidement déterminer l'accroissement de vitesse compte tenu de la perte de masse liée à la combustion des ergols.

0.2 Énoncé

On considère un lanceur spatial au décollage depuis la terre ; on demande de calculer la vitesse de ce lanceur en fonction du temps.

L'équation de Tsiolkovski est donnée par :

$$\delta V = V_e \cdot ln(\frac{M_0}{M_1})$$

où :

- v_e la vitesse d'éjection des gaz de la tuyère,
- M_0 la masse de la fusée à l'instant T_0,
- M_1 la masse de la fusée à l'instant T_1.

À chaque intervalle de temps Δt, la masse de la fusée est diminuée d'une certaine quantité de masse. Ceci permet de calculer l'accroissement de la vitesse de la fusée.

Nous considérons un lanceur de type Ariane 5, propulsé par un moteur Vulcain central et deux propulseurs à poudre latéraux. Les caractéristiques sont les suivantes :

- Masse de la fusée : 750 tonnes,
- Masse à vide (sans ergols) : 90 tonnes,
- V_e : 4 km/s,
- Perte de masse : 4 tonnes/s.

Les effets aérodynamiques sont négligés pour le calcul.

0.3 Travail à réaliser

On vous demande de réaliser un programme qui calcule la vitesse du lanceur en fonction du temps.

1. Concevoir le programme et décrivez l'algorithme (pseudo-code...).
2. Écrire un programme complet qui calcule et affiche en mètres par seconde de manière *itérative* par pas de 1 seconde la vitesse de la fusée depuis l'instant de départ au sol jusqu'à l'instant pour lequel la vitesse n'évolue plus.
3. Calcul de l'altitude du lanceur à chaque pas de calcul.

0.4 Objectifs pédagogiques

Les objectifs pédagogiques de cet exercice sont les suivants :

— compréhension d'un problème algorithmique en lien avec le monde réel,
— sélection de la bonne structure de contrôle ad hoc (boucles) pour résoudre le problème,
— élaboration de la condition d'arrêt de la boucle,
— itération par pas de 1 seconde,
— utilisation de constantes symboliques,
— calculs simples avec différentes unités.

```c
#include <stdio.h>
#include <math.h>

typedef struct Rocket {
    union {double f;  double fuel_mass;}; // tonnes
    union {double m;  double rocket_mass;}; // tonnes
    union {double ve; double ejection_speed;}; // m/s
    union {double me; double ejection_mass;}; // tonnes/s
    union {double h;  double altitude;}; // m
    union {double v;  double velocity;}; // m/s
} Rocket;

struct Time {int m; int s;};

struct Time s2ms(double s) {
    return (struct Time){.m = s / 60, .s = s - (int)(s / 60) * 60};
}

void display_time(struct Time t) {
    printf("%2d:%02d", t.m, t.s);
}

void display_rocket(Rocket r) {
    printf("Fuel mass: %.2f t\t", r.fuel_mass);
#ifdef VERBOSE
    printf("Rocket mass: %.2f\t", r.rocket_mass);
    printf("Ejection speed: %.2f\t", r.ejection_speed / 1000);
    printf("Ejection mass: %.2f\t", r.ejection_mass);
#endif
    printf("Altitude: %.2f km\t", r.altitude / 1000);
    printf("Velocity: %.2f km/s\t", r.velocity / 1000);
    printf("\n");
}

int main() {
    const double km = 1000;
    Rocket r = {
        .rocket_mass = 90, // tonnes
        .fuel_mass = 750 - 90, // tonnes
        .ejection_speed = 4 * km, // m/s
        .ejection_mass = 4, // tonnes/s
        .altitude = 0, // m
        .velocity = 0 // m/s
    };

    const double dt = 10.0;

    double t = 0;
    while(r.fuel_mass >= 0) {
        display_time(s2ms(t)); putchar('\t');
        display_rocket(r);

        double new_fuel_mass = r.f - r.me * dt;
        r.velocity += r.ve * log((r.m + r.f) / (r.m + new_fuel_mass));
        r.altitude += r.v * dt;
        r.f = new_fuel_mass;
        t += dt;
    };
}
```