

# TP 1 - Maîtriser les fondamentaux

## Structures de contrôle, corriger et dérouler un code

L2 SPI - UE Langage et programmation

Université d'Evry Val d'Essonne

---

### Exercice 1 : Utiliser les structures de contrôle

1. Ecrire un programme qui détermine si une année donnée par l'utilisateur est une année bissextile. On utilisera les propriétés suivantes :
  - Année Bissextile :
    - Si une année est divisible par 4, 100 et 400, il s'agit d'une année bissextile.
    - Si une année est divisible par 4 mais pas par 100, il s'agit d'une année bissextile.
  - Année non bissextile :
    - Si une année n'est pas divisible par 4, il ne s'agit pas d'une année bissextile.
    - Si une année est divisible par 4 et 100 mais pas par 400, il ne s'agit pas d'une année bissextile.
2. Modifier votre programme précédent pour que l'utilisateur puisse avoir la possibilité d'itérer sa demande. Le programme s'arrêtera lorsque l'utilisateur entrera la valeur -1 en guise d'année.
  - en utilisant **while**,
  - en utilisant **do - while**,Comparer et commenter les deux implémentations.
3. Modifier votre programme précédent pour qu'il puisse vérifier systématiquement si les années comprises entre 1971 et 2021 sont bissextiles. On utilisera la structure de contrôle **for**.
4. Ecrire un programme qui calcule le jour de la semaine qui correspond à une date donnée, cette dernière étant exprimée sous forme de trois nombres j (jour), m (mois) et a (année). On supposera que la date donnée est correcte. On utilisera les relations suivantes :

$$m_l = \begin{cases} m - 2 & \text{si } m \geq 3 \\ m + 10 & \text{si } m < 3. \end{cases}$$

$$a_l = \begin{cases} a & \text{si } m \geq 3 \\ a - 1 & \text{si } m < 3. \end{cases}$$

$n_s$  : deux premiers chiffres de  $a_l$

$a_s$  : deux derniers chiffres de  $a_l$

$$f = j + a_s + \frac{a_s}{4} - 2 \times n_s + \frac{n_s}{4} + \frac{26 \times m_l - 2}{10}$$

Dans ces conditions, le jour de la semaine est donné par le reste de la division de  $f$  par 7 (0=dimanche, 1=lundi, etc.). Attention :  $f$  peut avoir une valeur négative, ajouter la valeur 7 à  $f$  dans ce cas. On veillera à utiliser la structure de contrôle **switch-case**.

### Exercice 2 : Corriger les erreurs et dérouler un code

On se propose de réaliser un programme permettant de simuler le tir d'un canon. Nous pouvons contrôler l'angle de tir  $\alpha$ . La vitesse initiale à la sortie du canon est toujours la même  $V_0$  (voir figure).

On négligera les forces de frottement de l'air sur l'obus. Les équations cinématiques du mouvement de l'obus sur les axes  $x$  et  $y$  sont ( $g$  étant l'accélération de la pesanteur sur terre que l'on fixera à environ  $10\text{m/s}^2$ ) :

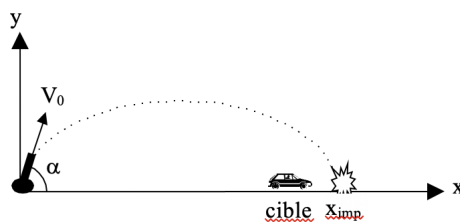
$$\begin{cases} y = -\frac{1}{2}gt^2 + V_0 \sin(\alpha)t \\ x = V_0 \cos(\alpha)t \end{cases}$$

On en déduit que :

$$y = F(x) = -\frac{g}{2V_0^2 \cos^2(\alpha)}x^2 + \tan(\alpha)x$$

Le point d'impact de l'obus sur le sol est à  $x_{imp}$  tel que  $F(x_{imp}) = 0$  avec :

$$x_{imp} = \frac{2V_0^2 \sin(\alpha) \cos(\alpha)}{g} = \frac{V_0^2 \sin(2\alpha)}{g}$$



1. Analyser le programme donné en annexe qui permet de calculer et de retourner la valeur  $x_{imp}$  du point d'impact de l'obus. Corriger dans un premier temps les erreurs de compilation. Pour chaque erreur relevée, compléter un tableau, en indiquant le numéro de la ligne et l'erreur identifiée, la manière dont vous la corrigez.
2. Dans un deuxième temps, déterminer si la fonction répond au cahier des charges. Pour cela, on s'assurera que la fonction retourne bien la bonne valeur de l'abscisse du point d'impact en posant par exemple  $V_0 = 2000\text{m/s}$  et en prenant successivement  $\alpha = \pi/4, \pi/6, \pi/3$ . Si le résultat donné par le programme ne correspond pas à la valeur théorique attendue, proposer les corrections nécessaires afin que le programme puisse réaliser les tâches pour lesquelles il est prévu.
3. Dérouler pas à pas votre code en complétant un tableau dans lequel vous noterez, pour toutes les variables utilisées, le numéro de la ligne et la valeur de la variable (état de la variable) à chaque fois que celle-ci est modifiée.
4. **[BONUS]** On souhaite réaliser un programme "de jeu" permettant à un utilisateur de choisir l'angle de tir (en degrés) puis de tirer un obus sur une cible immobile située à une distance **distanceCible** (choisie aléatoirement, mais à portée de tir). Lorsque l'obus tombe à une distance de la cible inférieure au rayon de destruction de l'obus **rayonDestruction** (considéré ici comme un indicateur de précision du tir), on considérera que la cible est touchée. Sinon, on affichera un message indiquant si le tir a été trop court ou trop long et l'utilisateur tentera un nouveau tir. Le programme se termine lorsqu'on a atteint la cible ("Bravo!").

Note : Évitez de vous trouver dans le rayon de destruction ou de tirer derrière vous !

ANNEXE

---

## Code à corriger, exercice 2, question 1 et 2.

---

```
1  #include <stdio.h>
2
3  int main()
4  {
5
6      #define G 10;
7
8      int V 0;
9      int langle;
10     int x imp;
11
12     printf(Donner la valeur de la vitesse initiale : );
13     scanf("%d", V 0);
14     printf(Donner la valeur de l'angle alpha en degres : );
15     scanf("%d", langle);
16
17     x imp = PointImpact(V 0, langle);
18     printf(Voici la valeur de l'abscisse du point d'impact : \n, x imp);
19
20 }
21
22 void PointImpact(int v0, int a)
23 {
24     int x;
25     x = 1/G* V 0* V 0 * sin[2*a]
26     return x;
27 }
```

---