

Outils pour l'exploitation des fichiers IGC

L'objectif de ce TD est de développer un outil qui lit et exploite des enregistrements de vols (format IGC). Il calcule les vitesses horizontales et verticales moyennes et enregistre les résultats dans un autre *fichier*.

Il est conseillé de lire tout l'énoncé avant de vous lancer

Le format IGC

Le format *IGC* est utilisé pour enregistrer les vols d'aéronefs, en particuliers ceux de vol libre (planeurs, delta, parapente), notamment lors des compétitions. C'est un format libre dont les spécifications sont publiées par *l'International Gliding Commission*.

Les fichiers *IGC* contiennent un grand nombre d'informations (identité du pilote, type d'épreuve, type d'aéronef, position, etc.) enregistrées sous forme **d'enregistrements** au format texte.

Il y a un enregistrement par ligne. Les premiers caractères donnent le type de donnée de l'enregistrement. Les enregistrements qui nous intéressent sont ceux de type *B*, donnant la position en 3D. En voici un exemple avec la signification à la suite.

Format d'un enregistrement de position (type B):

Voici un extrait de fichier *.igc* :

```
AXFI769302
HFDTE DATE:100622
HFPLTPILOTINCHARGE:Mathilde
(...)
LPLTFlymasterDesigner 2.04e
B1341234531771N00615152EA0119001285
B1341244531769N00615148EA0119001285
B1341254531767N00615146EA0119101285
B1341264531766N00615142EA0119101286
B1341274531765N00615138EA0119101286
B1341284531764N00615134EA0119101288
B1341294531762N00615131EA0119201289
B1341304531761N00615130EA0119401290
(...)
```

Prenons l'Exemple de l'enregistrement : B1341264531766N00615142EA0119101286

Il s'interprète comme suit :

- **B** : type de l'enregistrement. Ici celui qui nous intéresse.
- 134126 : horodatage. le point est fait à 13h41mn26s
- 4531766N : latitude. ici 45 degrés 31.766 minutes Nord
- 00615142E : longitude. ici 006 degrés 15.142 minutes Est
- **A** : validité de l'altitude. Le système d'enregistrement a considéré que l'altitude mesurée était valide
- 01191 : altitude barométrique. ici 1191m
- 01286 : altitude GPS. ici 1286m.

Les autres types d'enregistrements (entêtes != 'B') ne sont pas à traiter. Vous n'aurez qu'à les ignorer.

Enoncé

Sur la base d'enregistrements lus dans un fichier et donnés sous forme de chaînes, vous devrez implémenter des fonctions pour permettre d'extraire les informations de cette chaîne (date et position) et de les exploiter pour calculer des distances, vitesses, durées.

La lecture et l'écriture dans les fichiers est déjà codée. Elle vous crée un tableau de chaînes de caractères correspondant à chaque ligne du fichier lu.

Code mis à disposition

structures de stockage

On définit les structures suivantes pour enregistrer et manipuler les données qui nous intéressent. (cf. *tools.h*)

```
// enregistrement d'une position
typedef struct _record {
    char time[6];    // horodatage au format hhmmss
    float latitude;  // en degrés. Positive au nord
    float longitude; // en degrés. Positive à l'est
    int altitudeBaro; // altitude barométrique en mètres
    int altitudeGPS;  // altitude GPS en mètres
} IGCRecord;

// structure pour stocker les écarts entre 2 positions.
typedef struct _deltaRecord {
    char time[6];    // horodatage au format hhmmss
    int duree;       // écart temporel en secondes
    float distH;     // distance horizontale en m
    int distV;       // distance verticale en m. positif vers le haut ; négatif vers
```

```

float vitesseH;    //vitesse horizontale en m/s
float vitesseV;    //vitesse verticale en m/s. positif vers le haut ; négati
} IGCDeltaRecord;

```

constantes

On définit les deux constantes suivantes à utiliser pour représenter des informations non pertinentes ou non initialisées. (à vous d'en faire bon usage).

```

const IGCRecord NULLRecord; // l'enregistrement NULL.

const IGCDeltaRecord NULLDeltaRecord; // l'enregistrement NULL. idem NULLRecord.

```

On pourra par exemple s'en servir pour vérifier si une donnée est valide avant de l'exploiter :

```
if (data != NULLRecord) ...
```

Fonctions à coder dans *tools.c*

Vous écrirez les fonctions suivantes dans le fichier *tools.c*. Elles sont déclarées dans *tools.h*.

1. Une fonction qui revoie les informations d'un point à partir du tableau de char contenant un enregistrement. Si l'enregistrement n'est pas de type B ou qu'il est mal formaté, la fonction retourne *NULLRecord*.

```
IGCRecord extractIGC(char chaineEnregistrement[]);
```

2. Une fonction qui calcule la durée entre deux enregistrements ainsi que les distances et vitesses *verticales* et *horizontales* correspondantes.
On choisit de conserver l'horodatage de la position de départ.

```
IGCDeltaRecord calculerEcart(IGCRecord depart, IGCRecord arrivee);
```

3. Une fonction qui intègre plusieurs échantillons de *IGCDeltaRecord* en un seul comme suit :

- time : horodatage du premier échantillon
- durée = durée totale (i.e. somme des durée des échantillons)
- distance horizontale totale : somme des distances élémentaires
- distance verticale cumulée : somme **algébrique** des distances élémentaires. Ex. 3m positifs + 2m négatifs donne un cumule de 1m positif.
- vitesses : les vitesses **moyennes** sur la durée de l'échantillon. Pour la vitesse verticale

```
IGCDeltaRecord cumuleRecords(IGCDeltaRecord []);
```

Attention : pour utiliser cette fonction vous penserez à passer un tableau assez grand.

25/11/2022 17:13

Pour vos mises au point, vous utiliserez en entrée le fichier *exemple.igc* qui est un extract du gros fichier. L'exécution se fait avec la commande suivante : `./test.exe < exemple.igc > essai.csv` Votre résultat est stocké dans `essai.csv`. Vous pourrez le vérifier via un éditeur de texte.

Le fichier de test final est un enregistrement de vol réel (*20bf1b30-2b33-4174-87aa-abe78cf1db7c.igc* récupéré sur internet). La commande est : `./test.exe < 20bf1b30-2b33-4174-87aa-abe78cf1db7c.igc > AnalyseVol.csv`

Quelques recommandations

1. ne restez pas bloqué sur une fonction, vous pourrez y revenir plus tard.
2. le code a été compilé et testé (avant de mettre les trous bien évidemment). Il se peut tout de même qu'il y ait des erreurs. Dans ce cas, essayez de comprendre ce que l'on attend de votre programme et soyez inventifs.
3. soyez curieux. Notamment, si vous vous sentez, vous pouvez faire des moyennes glissantes ou d'autres analyses qui vous passent par la tête.