

# Algorithmique

## Et langage C

EPSI - B1 - DEVE102

Sujet du projet en binôme - Décembre 2018

Christopher Belliard - [christopher@coddity.com](mailto:christopher@coddity.com)

## Introduction

L'objectif de ce projet est de vous amener à travailler en binôme sur un cas concret et complet.

### Résultat attendu

- Code source documenté
  - Répertoire projet sans les sous-répertoires `bin` et `obj`
- Dossier projet :
  - Sommaire ;
  - Présentation de l'équipe ;
  - Présentation générale de votre programme ;
  - Algorithme schématique du fonctionnement du programme ;
  - Architecture technique : arborescence des fichiers, choix des structures de données, justification des variables principales, nomenclature des fonctions (prototype + rôle) ;
  - Documentation utilisateur pour la prise en main ;
  - Conclusion, auto-critique et perspectives potentielles d'évolution.

### Modalités de livraison

- Par mail à [christopher@coddity.com](mailto:christopher@coddity.com) dans un fichier compressé ;
- Au plus tard le jeudi 20/12/2018 à 13h37 ;
- Pénalité d'un demi-point toutes les 5 minutes de retard.

### Critères d'évaluation

- Correct : code compile, sans warnings, programme réalise ce qu'on attend de lui, pas d'anomalies ;
- Optimisé : nombre et taille des variables, quantité d'instructions, efficience algorithmique ;
- UX : beauté des écrans, soin apporté aux affichages ;
- Lisible : code clair, nommage des variables intuitif, commentaires fournis ;
- Qualité du dossier : exhaustivité, rédaction / orthographe.

# PARIS X GAMES 2019

Votre programme doit permettre à Bernard Laporte, président de la station de ski des Buttes Chaumont, d'organiser une compétition internationale : les Paris X-Games 2019.

Bernard attend de son programme un certain nombre de fonctionnalités. Il a fait un appel d'offres que vous avez remporté grâce à votre talentueux développeur commercial. Maintenant, il faut livrer.

Voici les spécifications fonctionnelles de l'application, rédigées de la main même de M. Laporte.

## Étape 0 : Accueil

Le programme devra afficher un écran d'accueil. M. Laporte n'a pas de souhaits particuliers mais il est très sensible aux belles choses. Vous avez donc carte blanche, soyez à la hauteur de votre réputation.

## Étape 1 : Inscriptions

La première étape consiste en l'enregistrement des skieurs participants. Les Paris X-Games 2019 étant une compétition de très haut niveau, le nombre maximum de participants a été établi à **20**.

Il est à noter qu'un skieur est identifié par son nom, sa nationalité et son numéro de dossard.

Les quatre nationalités pouvant concourir aux X-Games 2019 sont les suivantes :

Code	Nationalité
FR	Français(e)
IT	Italien(ne)
GE	Allemand(e)
NO	Norvégien(ne)

Le programme affichera le premier menu suivant :

- 1- Inscription d'un skieur
- \*- Etape suivante

L'option 1 permettra, comme son nom l'indique, d'inscrire un skieur. Pour chaque nouveau skieur :

- L'utilisateur saisira le nom (chaîne de caractères),
- Les nationalités et leur code s'afficheront,
- L'utilisateur saisira un code nationalité,
- Le programme attribuera un numéro de dossard automatiquement (en commençant à 1).

Règles de gestion :

- Si le code nationalité saisi n'existe pas, l'utilisateur devra saisir un nouveau code ;
- S'il s'avère que le skieur saisi est déjà inscrit (même nom et même nationalité), le programme devra avertir l'utilisateur et reprendre l'inscription de ce dernier au début.
- Le programme ne pourra pas enregistrer plus de 20 skieurs.

La saisie de la touche '\*' passera à l'étape suivante (applicable pour la suite du programme).

## Étape 2 : Récapitulatif des inscrits

Le programme proposera ensuite de récapituler les inscrits grâce à un menu à 3 entrées :

- 1- Par ordre d'inscription
- 2- Par nationalité
- 3- Modifier une inscription
- \*- Etape suivante

L'option 1 affichera la liste des skieurs (dossard - nom - nationalité) dans l'ordre d'inscription.

L'option 2 affichera, comme pour l'inscription, la liste des nationalités disponibles et demandera la saisie d'un code. Il affichera ensuite la liste des inscrits ayant cette nationalité sous le même format que l'option 1, dans l'ordre d'inscription également.

L'option 3 demandera de saisir un numéro de dossard et demandera de saisir à nouveau l'intégralité des infos du skieur concerné.

## Étape 3 : Descente des skieurs

Le programme proposera à l'utilisateur de saisir le temps de descente de tous les skieurs.

Il demandera un numéro de dossard, puis enregistrera le temps de descente du skieur correspondant. Le temps de descente sera saisi en secondes, avec les dixièmes et centièmes de seconde.

Exemple : 82.51 correspondra à 1 minute, 22 secondes et 51 centièmes.

Cas particuliers :

- Si le numéro de dossard saisi n'existe pas, le programme demandera une nouvelle saisie.
- En cas d'erreur sur le parcours (skieur disqualifié), l'utilisateur devra saisir 0.
- Un skieur ne s'étant pas présenté à la compétition est forfait : son temps de parcours ne doit pas être saisi du tout.

## Étape 4 : Résultats

Pour finir, le programme présentera un menu permettant d'afficher les résultats de la compétition :

- 1- Classement final
- 2- Résultat par nationalité
- 3- Liste des skieurs disqualifiés
- 4- Liste des skieurs forfaits
- \*- Quitter le programme

Le classement final présentera les skieurs du plus rapide au plus lent, avec les informations suivantes : Position, numéro de dossard, nom, nationalité, temps de descente.

Le temps de descente devra être affiché dans un format pratique de type : --m --s --.

Cas particuliers :

- En cas de disqualification, affichez le temps de descente "Disq.".
- En cas de forfait, affichez le temps de descente "Forf.".

## Conseils divers

### Lisibilité et compréhension de votre code

- Facilitez le travail des relecteurs en expliquant au maximum les étapes de votre programme par des commentaires (caractères spéciaux // ou /\* \*/);
- Commentez vos fonctions de la sorte, par exemple :

```
/*
 * Fonction X réalisant le traitement XXX.
 * Elle prend 2 paramètres a et b de type entier, correspondant à X et Y.
 * Elle renvoie un entier qui est ZZZ.
 */
int fonctionX (int a, int b) {
    XXX
}
```

- Choisissez des noms de variables explicites ;
- Prenez soin d'indenter votre code (tabulations) correctement pour faciliter sa lecture ;
- Faites des dessins pour modéliser votre programme, notamment les appels de fonctions ;
- Faites relire votre code par quelqu'un d'autre pour vous assurer qu'il est compréhensible et lisible.

### Travail itératif

- Démarrez par la création de l'ossature générale de votre programme : l'affichage et les menus ;
- Créez des fonctions indépendantes pour chaque fonctionnalité mais ne les codez pas toutes dans l'immédiat. Vous y reviendrez et les développerez progressivement ;
- Travaillez par itérations : terminez une fonctionnalité avant d'en démarrer une autre.
- Attaquez-vous aux fonctionnalités que vous trouvez les plus accessibles. Par extension, laissez de côté les options les plus complexes pour plus tard.

### Travail collaboratif

- Pour accélérer votre travail en binôme, partagez votre code grâce à git sur un dépôt distant (gitlab, github, bitbucket, etc.).

### Gestion des erreurs

- Veillez à traiter toutes les erreurs de saisie possibles, dans les menus ou lors de l'entrée de noms, temps de parcours ou code nationalité.

### Documentation projet

- Ne négligez pas la documentation ! Elle est au moins aussi importante que le programme en lui-même. Idéalement, démarrez sa rédaction dès le début du projet.