

**PROJET LANGAGE C (classe 3)**  
**Généralités et calculs graphiques**

**Préambule**

Sur votre ordinateur, construire un répertoire "EXERCISES\_C", qui sera livré zippé sur MyGES. Dans ce répertoire, pour chaque exercice, créer un répertoire "exercice\_1", "exercice\_2" etc... où seront copiés les fichiers sources et les fichiers exécutables des exercices rendus. A chaque exercice où des réponses/explications doivent être données, doit correspondre un fichier texte (utiliser Wordpad, Word ou n'importe quel traitement de texte) nommé : exercice\_i.txt, comprenant les réponses, ainsi que les justifications. Ne pas oublier :

- d'écrire des commentaires et **de construire des menus systématiquement (il doit être possible de relancer le programme plusieurs fois sans avoir besoin de le terminer)**
- de préciser le(s) nom(s) de(s) l'étudiant(s) (dans les fichiers sources et sur le support rendu)
- de copier les fichiers **sources (extension .c ou .cc)** et **exécutables (extension .exe)** dans le bon dossier : **bien noter que si le fichier source est fourni seul, l'exercice sera noté 0**

Remarques importantes :

- il est possible de discuter un algorithme à plusieurs, mais pas de rendre un même fichier source,
- aucun dossier ne doit être rendu par e-mail, mais chargé sur MyGES,
- en cas d'incompréhension sur le sujet, passer me voir,
- **tout dossier rendu en retard sera noté 0**
- **tout dossier rendu dans lequel un exercice est copié sera noté 0 (y compris un seul de la liste)!**

**L'utilisation de tableaux et/ou de fonctions est interdite dans ce PROJET.**

**Exercice 1**

Construire un programme C permettant de simuler le déplacement d'une puce qui se déplace aléatoirement sur une ligne, en avant ou en arrière, par pas de 1 ou -1. Par exemple, si elle est à l'emplacement 0, elle peut sauter à l'emplacement 1 ou -1 ; si elle est à l'emplacement 2, elle peut sauter à l'emplacement 3 ou 1, etc.

Le programme doit fonctionner en partant de 0 vers le point d'arrivée 5.

\*\*\*\*\*-2\*\*\*\*\*-1\*\*\*\*\*0\*\*\*\*\*1\*\*\*\*\*2\*\*\*\*\*3\*\*\*\*\*4\*\*\*\*\*5\*\*\*\*\*6\*\*\*\*\*7



PS : un affichage en mode texte de ce déplacement serait un plus (utilisation de conio).

**Exercice 2**

Dans la convention collective de la société ESGI, le calcul des jours de congés dus annuellement s'effectue de la manière suivante :

- si un salarié travaille dans l'entreprise depuis moins d'un an, il a droit à deux jours de congés par mois de présence; sinon il a droit à 25 jours.

- s'il s'agit d'un cadre âgé d'au moins 35 ans et d'ancienneté supérieure à 3 ans, il lui est accordé 4 jours en plus des 25 jours normaux; s'il est âgé d'au moins 45 ans, d'ancienneté supérieure à 5 ans, il lui est accordé 7 jours en plus des 25 jours normaux,
- si le salarié est un agent de maîtrise, il a droit à 2 jours supplémentaires à partir de trois années d'ancienneté.

Construire le programme permettant :

- de calculer le nombre de jours de congés auxquels a droit un salarié quelconque,
- d'en calculer la valeur financière, en utilisant le fait que 25 jours de congés sont payés l'équivalent d'un salaire brut mensuel.

### **Exercice 3**

Construire un programme C permettant de déterminer aléatoirement *n* lettres au hasard (majuscules ou minuscules), prises entre deux bornes, et affichant au fur et à mesure un résumé de la séquence choisie.

Par exemple, si les bornes sont *x* et *Z*, si la séquence est *xyxxxZZ*, *x2yx3ZZ* est affiché.

Prévoir :

- d'initialiser les paramètres (*n*, bornes etc ...)
- d'afficher le résultat,
- de relancer le traitement (comme dans tous les autres exercices)

### **Exercice 4**

Construire un programme C permettant de saisir un nombre entier et d'afficher ce nombre en inversant chaque paire de chiffres. Si le nombre de chiffres est impair, le dernier chiffre ne sera pas modifié.

Par exemple, 12345 sera affiché sous la forme : 21435; 23 sera affiché sous la forme 32.

### **Exercice 5**

Construire un programme simulant un jeu de bowling simplifié en 3 carreaux, avec enregistrement du score.

Règle (Wikipédia)

Une partie de bowling compte dix carreaux (ou 10 frames). Chaque joueur lance deux boules à chaque carreau, sauf en cas d'abat (**strike**). Un abat consiste à faire tomber les dix quilles avec la première boule. La réserve (**spare**) consiste à faire tomber les dix quilles avec les deux tirs consécutifs du carreau.

Le calcul des points se fait de la façon suivante :

- un abat (strike) vaut 10 points auquel on rajoute le nombre de quilles abattues lors des deux lancers suivants,
- une réserve vaut 10 points auquel on rajoute le nombre de quilles abattues au lancer suivant
- un "trou" ou jeu ouvert vaut le nombre de quilles abattues

Attention, en cas d'abat au dixième lancer, au dixième lancer, deux lancers supplémentaires sont accordés. En cas de réalisation d'une réserve, un lancer supplémentaire est accordé.

Exemple

**Carreau 1   Carreau 2   Carreau 3**

10 (abat) 7/3 (réserve) 9 0 (trou)

Le score du premier lancer vaut :  $(10 + 7 + 3) = 20$

Le score du premier lancer vaut :  $(7 + 3 + 9) = 19$

Le score du troisième lancer vaut  $(9 + 0) = 9$

Le jeu a donc rapporté 48 points.

### **Exercice 6**

Il s'agit cette fois de construire un petit problème de recherche. L'objectif est d'arriver à faire fonctionner une bibliothèque graphique simple sous licence GNU, nommée EZ-draw (<https://pageperso.lif.univ-mrs.fr/~edouard.thiel/ez-draw/>). Un descriptif sommaire est fourni sur l'extranet.

$$y = \pm \sqrt{1 + \frac{x}{2}} (x \pm \sqrt{x^2 + 8})$$

En utilisant la documentation EZ-Draw fournie avec le sujet, construire un programme respectant les règles suivantes permettant :

- de tracer la fonction  $P_n(x)$  pour un  $n$  quelconque saisi,
- de tracer les axes,
- d'appuyer sur des touches à déterminer pour permettre de jouer sur les graduations ainsi que de zoomer,
- de relancer le programme autant de fois que nécessaire.
- avancé : de calculer les coordonnées d'un point  $M$  sélectionné à l'aide de la souris ou de le faire se déplacer sur la courbe