

Nom : .....

Prénom : .....

Note : ..... / 20

## ❑ Exercice 1 : Ligne de commande

1. Compléter le tableau suivant qui donne quelques commandes et leur signification

<code>cd ~</code>	Le répertoire courant devient le répertoire de l'utilisateur
<code>mkdir exo1</code>	Créer le répertoire <code>exo1</code>
<code>touch rep.txt</code>	Créer le fichier vide <code>rep.txt</code>
<code>chmod 640 rep.txt</code>	Les droits sur <code>rep.txt</code> deviennent <code>rw-r-----</code>
<code>chmod u+x rep.txt</code>	Ajouter le droit d'exécution pour l'utilisateur sur le fichier <code>rep.txt</code>
<code>echo Bonjour &gt; rep.txt</code>	Ecrire "Bonjour" dans le fichier <code>rep.txt</code>
<code>cp rep.txt ~/Sauvegardes</code>	Copier <code>rep.txt</code> dans le dossier <code>~/Sauvegardes</code>
<code>ls -a</code>	Lister les fichiers (y compris les fichiers cachés)

2. Donner la commande permettant de créer un lien physique et expliquer rapidement la différence avec un lien symbolique.

La commande permettant de créer un lien physique est `ln` et celle permettant de créer un lien symbolique est `ln -s`. Lors de la création d'un lien symbolique on obtient deux références vers le même fichier. Un lien symbolique par contre est un simple raccourci.

3. On souhaite déplacer tous les fichiers du répertoire courant ayant l'extension `.c` vers le dossier `SourcesC` qui se trouve dans le répertoire parent. Quelle commande faut-il écrire ?

```
mv *.c ../SourcesC
```

## ❑ Exercice 2 : Programmation en C

## ⚠ Attention

Pour cet exercice, vous travaillerez à partir des fichiers `ex2_question1.c` et `ex2_question2.c` qui se trouvent dans votre répertoire personnel du lycée. Ces fichiers seront automatiquement récupérés en fin d'évaluation, n'oubliez pas de sauvegarder régulièrement.

1. Calcul d'une somme
- Compléter le fichier `ex2_question1.c` en y écrivant le code d'une fonction `divisible` prenant en argument deux entiers  $n$  et  $p$  et qui renvoie `true` si  $p$  divise  $n$  et `false` sinon.
  - A l'aide de la fonction précédente, calculer la somme des entiers strictement inférieurs à 10 000 et divisible par 3 ou 7 et donner la réponse trouvée par votre programme : 21 426 429

2. Manipulation de tableaux

- Ecrire le fichier `ex2_question2.c` une fonction `etendue` qui prend en argument un tableau et sa taille et renvoie l'écart maximal entre deux éléments de ce tableau. Par exemple, sur le tableau `int ex[7] = {1, 5, 3, 0, -1, 4, 8 }`, la fonction `etendue` doit renvoyer 9.
- Créer un tableau `int un[100]` de taille 100 et à l'aide d'une boucle, l'initialiser avec les valeurs prises par la suite  $(u_n)_{n \in \mathbb{N}}$  de terme général  $u_n = n^2 - 133n + 3822$  pour  $n = 0 \dots 99$ . C'est à dire que `tab[i]` doit contenir la valeur de  $u_i$  (pour  $i \in \llbracket 0; 99 \rrbracket$ ), par exemple `tab[0]=3822`.
- Déterminer l'écart maximal entre deux éléments du tableau `tab` défini à la question précédente et donner la réponse trouvée par votre programme : 4 422

Programme pour la question 1

```
1  #include <stdbool.h>
2  #include <stdio.h>
3
4  // Ecrire ci-dessous la fonction divisible
5  bool divisible(int n, int p)
6  {
7      return (n%p==0);
8  }
9
10 int main()
11 {
12     const int seuil = 10000;
13     int somme = 0;
14     for (int i=1;i<seuil;i=i+1)
15     {
16         if (divisible(i,3) || divisible(i,7))
17         {
18             somme = somme + i;
19         }
20     }
21     printf("Résultat = %d\n",somme);
22     return 0;
23 }
```

Programme pour la question 2

```
1  #include <stdio.h>
2
3  int etendue(int tab[], int taille)
4  {
5      int tmin = tab[0];
6      int tmax = tab[0];
7      for (int i=0;i<taille;i=i+1)
8      {
9          if (tab[i]<tmin)
10         {
11             tmin = tab[i];
12         }
13         if (tab[i]>tmax)
14         {
15             tmax = tab[i];
16         }
17     }
18     return tmax-tmin;
19 }
20
21 int main()
22 {
23     int un[100];
24     for (int i=0;i<100;i++)
25     {
26         un[i] = i*i - 133*i + 3822;
27     }
28     printf("Réponse = %d\n",etendue(un,100));
29 }
```

```
29     return 0;  
30 }
```