PROJET INFORMATIQUE

STI2D – 2020/2021

Réalisation d'une application innovante

Dans le cadre du projet informatique, on souhaite réaliser une application innovante en c# permettant de proposer plusieurs réalisations.

Le travail demandé nécessite du travail personnel. 9 heures d'encadrement et de suivi de projet seront assurés en salle.

Ce projet sera à faire de manière **individuelle** (pas de binôme ni de groupe) et en **deux temps** : une réalisation en **LARP** sera à faire au début permettant de réfléchir et d'écrire les algorithmes nécessaires en pseudo-code ou en organigrammes suivie par la suite **d'une programmation en C#.**

On souhaite proposer à l'utilisateur une application à travers un menu lui offrant la possibilité de choisir l'une des opérations ci-dessous. Le contenu des différents choix possibles est décrit dans les parties A, B et C.

En choisissant le « 6 », on obtient le menu secondaire ci-dessous pour une meilleure lisibilité de la gestion de la partie B :

```
Veuillez saisir votre choix :

6


 *********  Partie B *********


1: Calcul de la date du lendemain
2: Calcul du nombre de jours entre deux dates
3: Vérifier si une année est Bissextile
4: Calculer la date de pâques (il faudra choisir la méthode de calcul par la suite)
5: Afficher le calendrier d'une année
6: Méthode innovante avce les Dates
Saisir votre choix svp :
```

Après chaque choix de la partie B, une proposition de rebasculer vers le menu principal doit être prévue pour permettre à l'utilisateur de revenir vers les deux autres parties (A et C).

Ecrire un programme proposant une série de choix à l'utilisateur sous forme d'un menu principal. L'utilisateur choisira l'une des propositions affichées parmi les 7 ce qui permettra l'exécution du programme associé. Un message d'erreur sera affiché si le choix n'appartient pas à l'ensemble des choix possibles. Le menu principal doit être réaffiché après chaque choix tant que l'utilisateur n'a pas sélectionné le choix d'arrêt (le 8ème du menu principal).

Travail à faire

On propose de décomposer ce travail en trois parties A, B et C ; décrites ci-dessous :

A. Un peu de Maths (10 pts)

Cette section décrit les différentes options à proposer lors du lancement de votre application. Elle est dédiée en grande partie à des raisonnements mathématiques.

1. Nombre d'Armstrong ?

Le nombre d'Armstrong est définie par un entier naturel égal à la somme des cubes des chiffres qui le composent.

Exemple :

$\boldsymbol{Armstrong(153) = 1^3 + 5^3 + 3^3 = 153}$ **→ 153 est bien un nombre d'Armstrong**

$\boldsymbol{Armstrong(210) = 2^3 + 1^3 + 0^3 = 11}$ **→ 210 n'est pas un nombre d'Armstrong**

Afficher tous les nombres d'Armstrong se trouvant dans l'intervalle [100, 500] ; en proposant **deux versions** :

1- Parcourir tous les nombres entre 100 et 500 par pas de 1 ;
2- Utiliser les notions : unité, dizaine et centaine.

2. Nombre réversible ?

Ecrire un programme qui permet de compter le nombre de nombres réversibles compris entre une *valeurMin* et une *valeurMax* que l'utilisateur va saisir.

Un nombre entier est dit réversible si la somme du nombre lui-même et son miroir donne un nombre composé uniquement de valeur impaire.

Exemple : 36 + 63 = 99.

36 est donc un nombre réversible.

Note : il faut écrire un algorithme permettant de calculer le miroir d'un nombre x.

3. Somme des diviseurs

Calcul de la somme des diviseurs d'un nombre entier positif. Ecrire un sous module prenant en entrée un nombre entier positif saisi par l'utilisateur, affiche l'ensemble de ses diviseurs et retourne la somme des diviseurs de ce nombre.

Exemple :

L'utilisateur donne la valeur 6.

L'algorithme va lui afficher ; la liste des diviseurs de 6 et leur somme :

Diviseurs : 1 ; 2 ; 3 ; 6

1+2+3+6=12

Lors de la traduction de ce sous module en c#, la signature de l'exercice sera :
static void SommeDiviseur() ; dont le corps va permettre de :

- Saisir un entier positif par l'utilisateur
- Afficher l'ensemble des diviseurs de ce nombre et d'afficher également la somme de ces derniers.

4. Découverte d'une année (nombre mystère)

Cette fonctionnalité permet de trouver un nombre. Le sous programme à écrire permet de :

- Générer un nombre entier compris entre 1 et 2018 et le stocker dans une variable *anneeCachee*.
- Demander à l'utilisateur de trouver cette année secrète comprise entre 1 et 2018. Pour cela, le programme demandera la saisie de nombres entiers successifs jusqu'à ce que l'utilisateur découvre l'année mystère. Pour chaque nombre essayé, le programme affichera une demande avec un message explicite. Il comparera l'année saisie à l'année cachée et affichera un message "trop grande" ou "trop petite" jusqu'à ce que l'année mystère soit trouvée ; le message sera alors "gagné" en précisant le nombre de coups joués.

Pour générer un nombre entier aléatoire en c#, on utilise le code ci-dessous :

```
Random generateur = new Random();

int nombreCache = generateur.Next(1, 101);
```

Proposer une variante à ce sous programme de découverte d'une année avec un nombre limité de coups à jouer.

5. Somme des chiffres d'une puissance :

Concevoir un algorithme qui demande à l'utilisateur de saisir deux nombres entiers **positifs** « n » et « m » puis calcule et affiche la somme des chiffres se trouvant dans n^m.

Par exemple :

pour n=2, m =15, on calcule d'abord 2^{15}

- 2^{15} = 32768
- puis on identifie et on somme au fur et à mesure les chiffres se trouvant dans le nombre 2^{15} ➔ 3 + 2 + 7 + 6 + 8 = 26
- Le résultat final = 26

B. Manipulation des dates (8 pts)

On souhaite réaliser une application permettant de manipuler des dates.

a. Demander à l'utilisateur de saisir une date sous la forme de trois entiers : jour, mois et annee. Vérifier si cette date est valide. Exemple :
   i. 25 7 1981 est une date valide
   ii. 31 2 1950 est une date erronée (mois de Février).
b. Calculer et afficher la date du lendemain en fonction d'une date saisie sous la forme de trois entiers. Exemple : 15 09 2020, Le jour suivant : 16 09 2020. Attention au lendemain où l'on change de mois et/ou d'année.
c. Vérifier si l'année d'une date est une année bissextile ou pas. Pour rappel, une année est dite bissextile si :
   i. Elle est divisible par 4 mais pas par 100
   ii. Ou elle est divisible par 400
d. Calculer la date de paques. Pour cela, il existe différentes méthodes pour effectuer ce calcul. Les trois méthodes principales sont :
   i. Algorithme de Gauss
   ii. Algorithme de Meeus
   iii. Algorithme de Conway

   Ces algorithmes sont disponibles sur les liens Wikipédia ci-dessous. L'utilisateur pourra choisir l'un des trois algorithmes pour connaitre la date de paques.
   Remarque : il existe des versions différentes en fonction des calendriers Grégorien et Julien.

   - Wikipédia Gauss
   - Wikipédia Meeus
   - Wikipédia Conway

e. Afficher le calendrier d'une année en saisissant deux entiers : une année et un numéro de jour correspondant au premier Janvier de cette année. Il faut penser à vérifier si l'année est bissextile. Exemple pour l'année 2020, on va saisir 2020 et le 2 (2 correspond au Mercredi associé au 01/01/2020).

```
Quelle année ?
2020
Quel est le jour du 1er janvier (0=lundi, 1=mardi, ...) ?
2
     Janvier
lu ma me je ve sa di
 .  .  1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31

    Février
lu ma me je ve sa di
 .  .  .  .  .  1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29

    Mars
lu ma me je ve sa di
 .  .  .  .  .  .  1
 2  3  4  5  6  7  8
 9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29
30 31

    Avril
lu ma me je ve sa di
 .  .  1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30

    Mai
lu ma me je ve sa di
 .  .  .  .  1  2  3
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30 31
```

```
    Juin
lu ma me je ve sa di
 1  2  3  4  5  6  7
 8  9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30

    Juillet
lu ma me je ve sa di
 .  .  1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31

    Aout
lu ma me je ve sa di
 .  .  .  .  .  1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
31

    Septembre
lu ma me je ve sa di
 .  1  2  3  4  5  6
 7  8  9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30

    Octobre
lu ma me je ve sa di
 .  .  .  1  2  3  4
 5  6  7  8  9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30 31

    Novembre
lu ma me je ve sa di
 .  .  .  .  .  .  1
 2  3  4  5  6  7  8
 9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29
30
```

f. **méthode innovante** : proposer un sous-programme innovant et surtout que vous proposez (attention au copier/coller) permettant de déterminer le jour de la semaine associée à une date saisie par l'utilisateur. Exemple : 18 6 2020, l'affichage attendu : Jeudi 18 Juin 2020.

C. Innovation (2pts)

Cette section va vous permettre de proposer un algorithme innovant et créatif à votre choix qui n'est pas lié à tous les algorithmes que vous avez pu faire dans les sections ci-dessus.

Structuration du code en C#

En attendant de démarrer le chapitre « Méthodes » (au second semestre), vous devez structurer votre programme C# comme suit :

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ProjetInformatique
{
    class Program
    {
        static void Question1()
        {
            // Votre code ici ...
        }
        static void Main(string[] args)
        {
            Question1();
            Console.ReadKey();
        }
    }
}
```

Ensuite pour chaque question vous utiliserez la même structure. Avec X le numéro de la question à réaliser. N'oubliez pas de modifier le programme principale (main) en conséquence.

```
static void QuestionX() {
    // Votre code ici ...
}
```

En LARP, vous pourrez avoir la même structuration en utilisant la notion de « sous-module ».

Consignes de livraison

- Ce travail est à réaliser en monôme.
- La **première partie** est à rendre sur **DevinciOnline** :
    - Au plus tard le **Samedi 14 Novembre 2020** à **21h.**
    - Vous déposerez **sur DevinciOnline** (aucun rendu par mail ne sera pris en compte) au **format .zip :**
        - Votre **dossier** contenant **l'ensemble des algorithmes** rédigés en pseudo-code ou organigramme en utilisant **LARP**
        - Un fichier ReadMe.txt
            - Ce fichier sera votre journal de bord. Vous indiquerez votre état d'avancement au fur et à mesure concernant la réalisation de votre projet (en indiquant les dates et le travail fait), les points bloquants, ce que vous n'avez réussi à faire etc.

- La **seconde partie** est à rendre sur **DevinciOnline** :
    - Au plus tard le **Dimanche 03 Janvier 2021 à 21h**
    - Vous déposerez **sur DevinciOnline** (aucun rendu par mail ne sera pris en compte) au **format .zip :**
        - Votre **dossier Visual Studio**
            - Le dossier doit inclure le fichier **.sln** et surtout le sous dossier avec le **fichier .cs**
        - Un fichier ReadMe.txt
            - Ce fichier sera votre journal de bord. Vous indiquerez votre état d'avancement au fur et à mesure concernant la réalisation de votre projet, les points bloquants, ce que vous n'avez réussi à faire etc.

    - Un programme ne compilant pas ou ne s'exécutant pas entraine une note de 00/20 pour la partie réalisation du projet.
    - L'ensemble des codes sera analysé à l'aide d'un **système anti-plagiat**. Un plagiat entrain une note de 00/20 au module.