LIFAP1 – CC mi-parcours – Séquence 1

Contrôle Continu (Durée totale : 1h)

Lundi 24 octobre 2022

NOM

……

PRENOM

……

Numéro Etudiant

……

Groupe

……

Recommandations : *Les documents, calculatrice, téléphone portable sont interdits. La qualité de l'écriture et de la présentation seront prises en compte dans la note finale. Vous veillerez à* ***respecter*** *les notations et les règles d'écriture des algorithmes vues en cours et en TD. Un soin tout particulier devra être apporté à l'écriture des entêtes des différents sous-programmes.*

Partie A – Langage C/C++ **(/14)**

Suite cyclique à partir d'un certain rang

La suite *u* est définie par $u_0=a$ et $u_{n+1}=f(u_n)$ avec :

1. a un entier naturel non nul
2. f la fonction qui associe, à un entier naturel, la somme des carrés des chiffres de son écriture en base 10 .

Par exemple en choisissant a=2585, on obtient la suite : 2585, 118, 66, 72, 53, 34, 25, 29, …
En effet 2586 donne $2^2+5^2+8^2+5^2=118$ et $1^2+1^2+8^2=66$

Au bout d'un certain nombre d'itérations, la suite commence un cycle, à partir de la valeur 4 :

4 → 16 → 37 → 58 → 89 → 145 → 42 → 20 →

1- Ecrire en C/C++ une **fonction** `saisie_positive` qui renvoie un entier `n` strictement positif choisi par l'utilisateur. La saisie devra être recommencée tant que la valeur proposée n'est pas strictement positive.

```
int saisie_valeur ()
{
  int val;
  do
  {
    cout<<"donnez une valeur strictement positive"<<endl;
    cin>>val;
  } while (val<=0);
}
```

2- Ecrire en C/C++ une **fonction** `somme_carres_chiffres` qui calcule et retourne la somme des carrés des chiffres qui constituent le nombre `n` passé en paramètre.
Exemple : `somme_carres_chiffres (2585)` retourne `118` ($2^2 + 5^2 + 8^2 + 5^2 = 118$).

```
int somme_carre_chiffres (int n)
{
   int som = 0, chiffre;
   while (n!=0)
   {
      chiffre = (n%10);
      som += chiffre*chiffre ;
      n/=10;
   }
   return som;
}
```

3- Transformez l'entête de la fonction précédente en une **procédure** qui "retournera" la même valeur.

```
void scc_proc (int n, int &scc)
```

4- Ecrire en C/C++ un sous-programme `devient_cyclique` qui retourne le nombre d'itérations nécessaires pour que la suite prenne la valeur 4 et remplit un tableau 1D avec toutes les valeurs intermédiaires de la suite ; la valeur `n` de départ sera passée en paramètre.
Exemple avec la valeur de départ 2585. Le tableau rempli est
`2585 | 118 | 66 | 72 | 53 | 34 | 25 | 29 | 85 | 89 | 145 | 42 | 20 | 4 |`
La valeur retournée est 13.

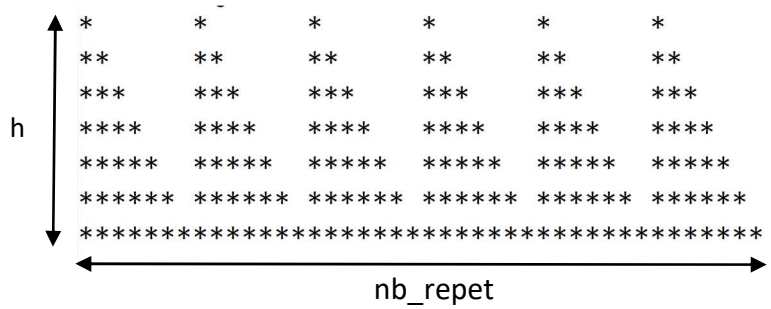
```
int tab_val (int T[MAX], int n)
{
   int iter = 0;
   T[0] = n;
   while (n != 4 )
   {
      iter ++;
      n =  somme_carre_chiffres(n);
     T[iter] = n;
   }
   return iter;
}
```

5- Ecrire en C/C++ le programme principal qui permet, en utilisant les sous-programmes précédents,
   a. de saisir un entier strictement positif,
   b. de remplir le tableau avec les différents termes de la suite jusqu'à la valeur 4,
   c. et d'afficher le nombre d'itérations nécessaires à la suite pour qu'elle atteigne à la valeur 4.

```
int main (void)
{
   int v = saisie_valeur();
   int T[MAX] = {0};
   int nb_iter = tab_val(T,v);
   cout<<"La suite devient cyclique a partir de "<<nb_iter<<" iterations"<<endl;
   return 0;
}
```

Partie B – Algorithmique (/6 pts)

On souhaite écrire un programme permettant d'affiche le motif suivant.



Ecrire l'algorithme d'un sous-programme `dessin` permettant d'afficher ce motif. La hauteur `h` et le nombre de répétitions du motif `nb_repet` sont passés en paramètres. Dans l'exemple h = 7 et nb_repet = 6.

```
Procedure dessin (h : entier, rep : entier)
Préconditions : h et rep > 0
Données : h, rep
Données / résultat : aucune
Description affiche le motif souhaité
Variables locales : i, j, l : entiers
Début
        Pour i allant de 1 à h par pas de 1 faire
                Pour l allant de 1 à rep par pas de 1 faire
                        Pour j allant de 1 à i par pas de 1 faire
                                Afficher ("*")
                        Fin pour
                        Pour j allant de 1 à h-i par pas de 1 faire
                                Afficher ("_")
                        Fin pour
                Fin pour
                Afficher saut de ligne
        Fin pour
Fin
```