

LIFAP1 – CC mi-parcours – Séquence 3

Contrôle Continu (Durée totale : 1h)

Mardi 25 octobre 2022

*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 – Algorithmique

On souhaite écrire un programme permettant d'afficher le motif ci-contre.

NOM
.....
PRENOM
.....
Numéro Etudiant
.....
Groupe
.....

- 1- Ecrire l'algorithme d'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.

fonction <code>saisie_valeur ()</code> : entier	*
Préconditions : aucune	*
Données : aucune	**
Données / résultat : aucune	*
Résultat : entier	**
Description saisit et retourne une valeur strictement positive	***
Variables locales : <code>val</code> : entier	*
Début	**
Faire	***
Afficher("donnez une valeur strictement positive")	****
Saisir (<code>val</code>)	
Tant que (<code>val</code> <=0)	*
Retourner <code>val</code>	**
Fin	***

- 2-Ecrire l'algorithme d'un sous-programme `dessine_triangle` qui dessine **un** triangle d'étoiles de hauteur `h` passée en paramètre. On utilisera `afficher(saut de ligne)` pour passer à la ligne.

Procédure <code>triangle (h : entier)</code>
Préconditions : <code>h > 0</code>
Données : <code>h</code>
Données / résultat : aucune
Description affiche le motif souhaité
Variables locales : <code>i, j</code> : entiers
Début
Pour <code>i</code> allant de 1 à <code>h</code> par pas de 1 faire
Pour <code>j</code> allant de 1 à <code>i</code> par pas de 1 faire
Afficher ("*")
Fin pour
Afficher saut de ligne
Fin pour
Fin

- 3- Ecrire l'algorithme d'un sous-programme `dessine_motif` qui dessine le motif complet de hauteur h . Il y aura un triangle de hauteur 1, un triangle de hauteur 2, ... un triangle de hauteur h .

```
Procédure motif (h : entier)
Préconditions : h > 0
Données : h
Données / résultat : aucune
Description affiche le motif souhaité
Variables locales : i : entiers
Début
    Pour i allant de 1 à h par pas de 1 faire
        Triangle(i)
        Afficher (saut de ligne)
    Fin pour
Fin
```

- 4- Ecrire l'algorithme du programme principal qui en utilisant les sous-programmes précédents,
- saisit une valeur strictement positive,
 - dessine le motif souhaité.

```
Début
    Nb : entier
    Nb ← saisie_valeur()
    motif (Nb)
Fin
```

Partie B – Langage C/C++

Indice de masse corporelle

On imagine une visite médicale en deux parties où les patients se présentent d'abord tous pour la mesure de leur taille, puis repassent, dans le même ordre, à la pesée.

- 1- Ecrire en C/C++ une **fonction** `saisie_taille` qui remplit un tableau 1D de taille `MAX` (constante définie au préalable) avec les tailles des différents patients. La taille sera saisie par l'utilisateur et cette saisie s'arrêtera dès qu'une valeur négative sera donnée. On renverra le nombre de valeurs présentes dans le tableau (nombre de patients). La taille sera donnée en mètres.

```

int saisie_taille (float T[MAX])
{
    int nb_patient = 0;
    do
    {
        cout<<"donnez la taille du patient "<<nb_patient<<" ou -1 pour stopper la saisie"<<endl;
        cin>>T[nb_patient];
        nb_patient++;
    } while (T[nb_patient-1]!=-1.0);
    return nb_patient-1;
}

```

- 2- Ecrire en C/C++ un sous-programme `saisie_poids` qui pour chaque patient précédemment mesuré, enregistre sa masse `m` en kg dans un tableau 1D de taille `MAX`. L'ordre de saisie sera identique pour les deux tableaux, et le nombre de patients `nb_patient` sera fourni en paramètre.

```

void saisie_poids (float T[MAX], int nb_patient)
{
    int i;
    for (i=0;i<nb_patient;i++)
    {
        cout<<"poids du patient "<<i<<endl;
        cin>>T[i];
    }
}

```

- 3- Ecrire en C/C++ un sous-programme `moyennes` qui calcule et "retourne" la taille moyenne et le poids moyen de tous les individus enregistrés dans les 2 tableaux précédents.

```
void moyennes (float Taille[MAX],float Poids[MAX],int nbp, float &taille_moyenne,float &poids_moyen)
{
    int i;

    poids_moyen = 0;
    taille_moyenne = 0;
    for (i=0;i<nbp; i++)
    {
        poids_moyen+=Poids[i];
        taille_moyenne+=Taille[i];
    }
    poids_moyen/=nbp;
    taille_moyenne/=nbp;
}
```

- 4- Ecrire en C/C++ le programme principal qui, en utilisant les sous-programmes précédent,
- saisit la taille d'autant de patients que l'utilisateur le voudra et les stocke dans un tableau,
 - saisit ensuite la masse de chacun d'entre eux dans un second tableau,
 - puis affiche la moyenne de la taille et du poids des patients enregistrés.

```
int main (void)
{
    float taille[MAX]={0};
    float poids[MAX]={0};
    int nbp=saisie_taille(taille);
    float pm, tm;
    saisie_poids(poids,nbp);
    moyennes(taille,poids,nbp,tm,pm);
    cout<<tm<<" "<<pm<<endl;
    return 0;
}
```