Cahier de TD n°4

Séance 1 : Définition de fonctions, vrai/faux

Quelques définitions

Parmi les entêtes qui suivent, lesquelles sont incorrectes et pourquoi ?

```
function 3x(reel x, reel y, reel z)

fonction racine_car(entree : reel → sortie : reel)

fonction racine(entree : reel toto → sortie : reel)

fonction foo(entree : entier a,b,c → sortie : entier,reel)

fonction bar(→ sortie : caractere)

fonction x3Y_fT2Ahahaha(entree : entier Azrz4_uho_hihihi)

fonction gauss(entree : reel x, entier sigma → sortie : reel)
```

VRAI/FAUX

- Un argument est une entrée pour la définition de la fonction;
- Un argument est une valeur donnée par un programme lors de l'appel d'une fonction;
- Une fonction a au moins une sortie;
- Une fonction a au plus une sortie;
- Il faut écrire autant d'exemplaires d'une fonction que de nombres de fois où on veut l'utiliser;
- Une fonction rend un programme plus lisible;
- Un programme avec des fonctions est plus dur à maintenir qu'un programme sans fonctions (maintenir = faire la chasse aux bugs);
- On doit obligatoirement appeler une fonction que l'on définit;
- On doit obligatoirement définit une fonction que l'on appelle;
- `afficher` est une fonction;
- `while` est une fonction.

fonctions simples, définitions et appels

exemples de définition et d'appels de fonction :

Ecrivez les programmes suivants avec des fonctions. Le programme principal devra appeler la ou les fonctions définies au moins 1 fois.

- Programme faisant la somme de deux entiers;
- Programme calculant la moyenne de deux entiers;
- Programme proposant un affichage propre : on fournit un nombre entier, par exemple qui indique le nombre N d'€ que vous avez gagné dans un jeu. Selon la valeur de N, qui peut être nulle, égale à 1 ou supérieure, le programme affichera un message tenant compte de cette valeur de N, notamment pour gérér le cas singulier/pluriel : on veut éviter d'afficher, par exemple, les messages :
  "Vous avez gagné 1 euros" (car on doit mettre euro au singulier ici).

arguments / paramètres / variables locales

Dans le programme suivant, indiquez où sont les arguments, les paramètres, les appels et les définitions des fonctions :

```
fonction gilb_1(entree : entier g1, entier g2 → sortie : entier)
{
      entier g_temp;

      g_temp ← g1+g2;
      g2 ← g_temp - g2;
      g1 ← g1 - g_temp + g1;
      g2 ← g1 - g2;

      retourner(g_temp);
}

fonction gilb_2(entree : reel g_1, caractere g_3 → sortie : reel)
{
      entier g_temp;

      g_temp ← g_1;
```

```
        retourner(g_1/g_temp);
}

programme gilb_fonc
{
        entier g_1, g_2, g_cpt;
        reel g_x;

        afficher("g_entrez deux g_entiers :");
        saisir(g_2);
        saisir(g_1);

        gilb_2(1.34543,'X');

        afficher("g_resultat :",gilb_1(g_2-g_1,g_1-g_2));

        g_1 ← gilb_1(g_1,g_2);
        afficher("g_resultat 2 : ",gilb_1(gilb_1(g_1,g_2),g_2));

        afficher("entrez le g_nombre de g_points :");
        saisir(g_1);

        g_x ← 0.0:

        pour g_cpt de 0 a g_1
        {
                g_x ← 1.0 + g_cpt * (5.0/g_1);
                afficher(g_x,"  ", gilb_2(g_x,'#'),"\n");
        }
}
```

Gilbert vous dit que toutes les variables sont locales. Etes-vous d'accord avec lui ? Que fait ce programme ? Quel serait l'allure de la courbe dessinée à partir des points calculés à la fin du programme ?

Séance 2 : Fonctions

Un programme mal nommé

Donnez les valeurs des variables au fur et à mesure du déroulement du programme suivant : que constatez-vous ?

```
fonction echange(entree : entier a, entier b)
{
        entier excg;
        excg ← a;
        a ← b;
```

```
    b ← exg;

    afficher("a = ",a," et b = ",b,"\n");
}

programme swap

entier a,b;

a ← 5;
b ← 2;
afficher("a = ",a," et b = ",b,"\n");
echange(b,a);
afficher("a = ",a," et b = ",b,"\n");
echange(a,b);
afficher("a = ",a," et b = ",b,"\n");
```

Fonctions utilitaires

Note : vous pourrez réutiliser ces fonctions à de nombreuses occasions pour vos projets futurs !

Ecrire une fonction qui affiche tous les éléments d'un tableau dont le type est connu (à vous de choisir le type). Précisez quelles sont les entrées et la sortie de cette fonction

Ecrire une fonction qui effectue la saisie d'un certain nombre d'éléments à ranger dans un tableau dont le type est connu. Précisez quelles sont les entrées et la sortie de cette fonction.

Rappelez pourquoi il n'est pas nécessaire de retourner un tableau dans une fonction qui modifie les éléments stockés dans le tableau.

Ecrire une fonction qui effectue le tri d'un tableau par ordre croissant ou décroissant, le choix de l'ordre de tri se fait par l'intermédiaire d'un paramètre `ordre_tri`. Précisez quelles sont les entrées et la sortie de cette fonction.

Retourner un tableau ?

Cas d'un tableau statique

Une fonction peut parfois avoir besoin de retourner un tableau, par exemple une fonction qui effectue une copie d'un tableau d'entiers, ou de réels. Que se passe-t-il exactement dans ce cas ? Nous allons tenter de le déterminer en regardant le programme suivant.

```
fonction copie_tab(entree : entier tab[], entier util → sortie :
entier [])
{
      entier tab_res[100];
      entier cpt;

      pour cpt de 0 à util-1
      {
            tab_res[cpt] ← tab[cpt];
      }

      retourner tab_res;
}

fonction affich_tab(entree : entier tab[], entier t_ut)
{
      entier cpt;

      pour cpt de 0 à t_ut-1
      {
            afficher(tab[cpt]," ");
      }
      afficher("\n");
}

fonction principale()
{
      entier tablo[100] ← {1,2,3,4,5,6,7,8};
      entier util;
      entier *resultat;

      util ← 8;

      resultat ← copie_tab(tablo, util);

      affich_tab(resultat, util);
}
```

question 1) Quel est le type de la sortie de la fonction `copie_tab` ?

question 2) Pourquoi doit-on utiliser un tableau dynamique pour la variable `resultat` du programme principal et non une définition statique telle que : `entier resultat[100]` ?

question 3) Que vaut, dans la fonction `copie_tab`, la variable `tab_res` ? (vous pouvez faire une hypothèse sur la valeur numérique de cette variable).

question 4) Que vaut, dans le programme principal, la variable `tab_res` ?

question 5) Que vaut la variable `resultat` du programme principal après l'appel à la fonction `copie_tab` ?

question 6) Que trouve-t-on en mémoire à l'adresse stockée dans la variable `resultat` du programme principal ?

question 7) Le programme affichera-t-il les valeurs du tableau ou affichera-t-il un message d'erreur ?

Séance 3 :

Cas d'un tableau dynamique

Voici une nouvelle version de la fonction copie_tab, fonctionnant avec un tableau dynamique. Répondez de nouveau aux questions 4 à 7 de l'exercice précédent, et indiquez les instructions à ajouter éventuellement dans et/ou à la fin du programme principal pour adapter le programme à la nouvelle version de cette fonction.

```
fonction copie_tab(entree : entier tab[], entier util → sortie :
entier *)
{
      entier *tab_res;
      entier cpt;

      tab_res ← reservation(util entier);

      si (tab_res ≠ NULL) alors
      {
            pour cpt de 0 à util-1
            {
                  tab_res[cpt] ← tab[cpt];
            }
      }

      retourner tab_res;
}
```

les paramètres de type pointeur

rappelez l'utilité des paramètres qui sont des pointeurs

Ecrire un programme avec une fonction qui réalise l'échange de 3 valeurs entières : la première reçoit la valeur de la deuxième, la deuxième celle de la troisième et enfin la troisième celle de la première. Cet échange doit être répercuté dans le programme principal.

Ecrire une fonction à qui l'on fournit un tableau statique et qui retourne un tableau dynamique contenant juste les éléments utilisés du tableau statique passé en paramètre.

Ecrire un programme appelant cette fonction.

Gestion des valeurs de retour : Les procédures

On appelle procédure une fonction dont la valeur de retour est un entier qui n'est pas le résultat d'un calcul. Cette valeur de retour est utilisée comme indicateur pour que le programme puisse savoir s'il peut se fier aux traitements que la fonction a effectués. Cela suppose également que la sortie de la fonction étant utilisée pour communiquer un entier, toute autre valeur que la fonction doit communiquer doit être gérée par un paramètre de type pointeur.

Ecrivez une fonction racine_carrée qui calcule la racine carrée d'un nombre positif. La valeur de retour de cette fonction n'est pas le résultat du calcul, mais indique si le calcul a pu être effectué ou non.

Ecrivez une fonction bruit qui reçoit un tableau de réel et calcule la moyenne, la médiane et l'écart-type de ces valeurs. Si l'écart-type des valeurs est inférieur à un seuil S fourni en paramètre, alors on considère que les calculs effectués ne sont pas fiables, sinon on considère que les résultats sont corrects.

Ecrivez une fonction sensée recevoir un tableau t de nombres tous positifs et qui effectue un calcul de moyenne pondérée suivant : m = (t[0]+2*t[1]+2*t[2]+t[3])/6.

Quelles sont les cas de figures dans lesquels la fonction ne pourra pas effectuer correctement ses calculs ? Vous en déduirez les codes d'erreur possibles pour cette fonction.

Ecrivez une deuxième fonction qui affiche un message d'erreur compréhensible en fonction de la valeur de retour de la fonction précédente.