

Second TP : pointeurs et tableaux statiques

Pendant la séance en leur présence, les enseignants de TP peuvent répondre à vos questions et vous assister, profitez-en donc pour envisager les différents aspects du contenu pédagogique (pointeurs, structures, tableaux, traductions en C++) pour la rendre profitable. Un travail préalable sur papier (ou, si vous en avez l'occasion, sur machine) est indispensable pour rentabiliser le temps de la séance.

1. Crazy

- Simuler l'exécution d'algorithme ci-dessous et écrire (sur papier) les états de chaque variable lors de l'exécution.
- Recopier l'algorithme en C++ et comparer la sortie avec votre réponse.

```
int main ( )
{
    int a = 1, b = 2, c = 3;
    int *p1, *p2;
    p1 = &a;
    p2 = &c;
    *p1 = ( *p2 )++;
    p1 = p2;
    p2 = &b;
    *p1 -= *p2;
    ++( *p2 );
    *p1 *= *p2;
    a = ( ++( *p2 ) ) * *p1;
    p1 = &a;
    *p2 = *p1 /= *p2 ;

    return EXIT_SUCCESS;
}
```

2. Personne

- Définir un type **t_date** pour définir une date
- Définir un type **t_personne** pour une personne, avec son nom, son prénom, sa date de naissance, son âge, et son adresse (utiliser string ou les tableaux de caractères).
- Définir une procédure permettant d'afficher l'information d'une personne passée en paramètre. Faut-il mieux passer un **t_personne** ou un **t_personne*** ? Ecrire les deux versions.
- Écrire une procédure permettant de lire les informations d'une personne saisies via le clavier. Ecrire une version procédure et une version fonction. Analyser les avantages et inconvénients de chaque approche.
- Ecrire les procédures d'affichage et de saisie de **t_personne** en utilisant les références.
- Tester vos fonctions dans un programme principal.

3. Binome

- Définir une structure **t_binôme** pour définir un binôme entre deux personnes. Allez-vous utiliser les attributs de type **t_personne** ou **t_personne*** ? Analyser chaque approche.
- Définir une procédure permettant d'afficher l'information sur un binôme, considérer l'écriture de deux fonctions : affichage complet ou succinct. Considerer des différents scénarios: le binôme n'est pas défini, incomplet, etc.
- Ecrire une procédure permettant de créer un binôme à partir de deux variables **t_personne** préexistantes. Puis, écrire une version fonction en utilisant les pointeurs et les références.
- Tester vos fonctions dans un programme principal. Attention, testez tous les scénarios possibles !

4. Loto(*)

- Écrire un programme qui déclare un tableau de 6 entiers, et les saisit au clavier. Ces entiers vont correspondre à un tirage de loto.
- Proposer et tester une fonction **EstPresent** qui accepte en entrée deux paramètres, une valeur et un tableau de 6 entiers, et retourne
 - ☐ 1 si la valeur est présente dans le tableau
 - ☐ et 0 si la valeur n'apparaît pas
- En déduire un programme qui, en plus des six numéros du tirage du loto, demande les 6 numéros d'une grille de loto, et affiche les numéros qui sont présents dans le tirage. À la fin, le programme affiche le nombre de bons numéros (avec un texte particulier si on a les 6 bons numéros).
- Ré-écrire la fonction **EstPresent** en procédure (il n'y a plus de valeur renvoyée)
- Écrire une procédure **SaisieGrille** qui effectue la saisie d'une grille. Pour chaque numéro saisi, on vérifiera que la valeur est correcte (entre 1 et 49, et différente des valeurs précédemment saisies), sinon, on demandera de la re-saisir. Intégrer cette fonction dans le programme pour la saisie du tirage et d'une grille.