

Séance 20 : Jeudi 02/05/2019

Terenui est un véritable Globetrotter ... Il parcourt le monde à la recherche des paysages les plus idylliques.

Aussi afin de conserver une trace de ses aventures, nous avons décidé de conserver une trace de ses différents périples.

Pour chaque voyage, on renseignera les informations suivantes :

- Destination
- Continent (Asie, Afrique, Europe, Amérique, Océanie)
- Durée du séjour (en nombre de jours)
- Appréciation (note sur 20)

Sachant qu'il effectue **au plus** 20 voyages par an, nous allons définir une **structure** contenant le nombre de voyages déjà effectués ainsi qu'un tableau de voyages (caractéristiques de chacun des voyages effectués).

1. Définir en C/C++ les structures de données nécessaires à la mise en œuvre du programme : structure voyage et structure tous_les_voyages.
2. Écrire en C/C++ un sous-programme remplir_voyage permettant de saisir **toutes** les informations relatives à un voyage.
3. Écrire en C/C++ un sous-programme afficher_voyage permettant de d'afficher **toutes** les informations relatives à un voyage.
4. Écrire en C/C++ un sous-programme ajoute_voyage permettant d'ajouter les caractéristiques d'un voyage au tableau de voyages. **On utilisera pour cela le sous-programme écrit en 2.**
5. Écrire en C/C++ un sous-programme moyenne_par_continent qui calcule et "renvoie" au programme principal la moyenne des appréciations des voyages effectués **pour chacun** des 5 continents. On utilisera un tableau tab_moyenne pour stocker ces 5 valeurs (indice 0 : Afrique, indice 1 : Amérique, indice 2 : Asie, indice 3 : Europe, indice 4 : Océanie).
On supposera que chaque continent aura été visité au moins une fois.
6. Écrire en C/C++ un sous-programme le_mieux_et_le_pire permettant (à partir du tableau des moyennes défini en 4) de renvoyer l'indice du continent ayant la meilleure moyenne des appréciations, et l'indice du continent ayant la plus mauvaise moyenne.
7. Écrire en C/C++ le programme principal permettant d'ajouter dans le tableau de **voyages autant de caractéristiques de voyages que l'utilisateur le voudra** et d'afficher le **nom** du continent préféré de Terenui et celui qui l'enthousiasme le moins.

```

#include <iostream>
#include <string.h>
using namespace std;
const int MAX_VOYAGES = 20 ;
const int MAX_CH = 16 ; // constante utilisée pour les chaines de caractères
struct voyage{
    char dest[MAX_CH], continent[MAX_CH];
    int duree, note ;
};
struct tous_les_voyages{
    int nbvoyages;
    struct voyage tabvoyage[MAX_VOYAGES];
};
struct voyage remplir_voyage_fct (){
    struct voyage v ;
    cout << "Donnez la destination"<<endl;
    cin>>v.dest;
    cout << "Donnez le continent"<<endl;
    cin>>v.continent;
    do
    {
        cout << "Donnez la duree du sejour"<<endl;
        cin>>v.duree;
    } while (v.duree <=0);
    do
    {
        cout << "Donnez la note attribuee a ce sejour"<<endl;
        cin>>v.note;
    } while (v.note <0 || v.note>20);
    return v;
}
void remplir_voyage_proc (struct voyage &v){
    cout << "Donnez la destination"<<endl;
    cin>>v.dest;
    cout << "Donnez le continent"<<endl;
    cin>>v.continent;
    do
    {
        cout << "Donnez la duree du sejour"<<endl;
        cin>>v.duree;
    } while (v.duree <=0);
    do
    {
        cout << "Donnez la note attribuee a ce sejour"<<endl;
        cin>>v.note;
    } while (v.note <0 || v.note>20);
}
void afficher_voyage(struct voyage v){
    cout << "Destination : ";
    cout<<v.dest<<endl;
    cout << "Continent : ";
    cout<<v.continent<<endl;
    cout << "Duree du sejour : ";
    cout<<v.duree<<endl;
    cout << "Note attribuee a ce sejour : ";
    cout<<v.note<<endl;
}
void ajoute_voyage (struct tous_les_voyages &tlv){
    if (tlv.nbvoyages <MAX_VOYAGES) // on verifie qu'il reste de la place dans le
tableau
    {
        tlv.tabvoyage[tlv.nbvoyages] = remplir_voyage_fct();
        //remplir_voyage_proc(tlv.tabvoyage[tlv.nbvoyages])
        tlv.nbvoyages ++;
        cout<<"Il y a "<<tlv.nbvoyages<<" voyage(s) dans la structure"<<endl;
    }
}

```

```

    else cout<<"plus de place disponible ajout impossible"<<endl;
}
void moyenne_par_continent (struct tous_les_voyages tlv, float moyenne[5]){
    int i;
    int nb_voy_continent[5] = {0};
    for (i=0;i<5;i++)
    {
        moyenne[i] = 0;
    }
    for (i=0;i<tlv.nbvoyages;i++)
    {
        if (strcmp (tlv.tabvoyage[i].continent, "afrique")== 0)
        {
            moyenne[0] += tlv.tabvoyage[i].note;
            nb_voy_continent[0] ++;
        }
        if (strcmp (tlv.tabvoyage[i].continent, "amerique")== 0)
        {
            moyenne[1] += tlv.tabvoyage[i].note;
            nb_voy_continent[1] ++;
        }
        if (strcmp (tlv.tabvoyage[i].continent, "asie")== 0)
        {
            moyenne[2] += tlv.tabvoyage[i].note;
            nb_voy_continent[2] ++;
        }
        if (strcmp (tlv.tabvoyage[i].continent, "europe")== 0)
        {
            moyenne[3] += tlv.tabvoyage[i].note;
            nb_voy_continent[3] ++;
        }
        if (strcmp (tlv.tabvoyage[i].continent, "oceanie")== 0)
        {
            moyenne[4] += tlv.tabvoyage[i].note;
            nb_voy_continent[4] ++;
        }
    }
    for (i=0;i<5;i++)
    {
        moyenne[i] = moyenne[i] / nb_voy_continent[i];
    }
}
void affiche_tab_moyenne(float tab[5]){
    int i;
    cout<<"affichage du tableau de moyennes "<<endl;
    for (i=0;i<5;i++)
    {
        cout<<tab[i]<<" ";
    }
    cout<<endl;
}
void meilleur_pire (float moyenne[5], int &meilleur, int &pire){
    int i;
    meilleur = 0;
    pire = 0 ;
    for (i=1;i<5;i++)
    {
        if (moyenne[i]<moyenne[pire]) pire = i;
        if (moyenne[i]>moyenne[meilleur]) meilleur = i;
    }
}
int main (void)
{
    struct tous_les_voyages tlv1;
    float mpc[5];
    int best,worst;
    char rep ;
}

```

```

tlv1.nbvoyages = 0;
do
{
    ajoute_voyage(tlv1);
    cout<<"Encore un voyage ? (O/N) "<<endl;
    cin>>rep ;
} while (rep=='O' || rep=='o'); // on continue tant que l'utilisateur donne o
ou O comme réponse
moyenne_par_continent(tlv1,mpc);
affiche_tab_moyenne(mpc);
meilleur_pire(mpc,best,worst);
cout<<"le continent ayant la meilleure moyenne est : ";
switch (best)
{
    case 0 : cout<<"l'afrique"<<endl;break;
    case 1 : cout<<"l'amerique"<<endl;break;
    case 2 : cout<<"l'asie"<<endl;break;
    case 3 : cout<<"l'europe"<<endl;break;
    case 4 : cout<<"l'oceanie"<<endl;break;
    default : cout<<"erreur"<<endl;
}
cout<<"le continent ayant la pire moyenne est : ";
switch (worst)
{
    case 0 : cout<<"l'afrique"<<endl;break;
    case 1 : cout<<"l'amerique"<<endl;break;
    case 2 : cout<<"l'asie"<<endl;break;
    case 3 : cout<<"l'europe"<<endl;break;
    case 4 : cout<<"l'oceanie"<<endl;break;
    default : cout<<"erreur"<<endl;
}
return 0;
}

```