

1 Nom et prénom (Bases)

- Veuillez créer une fonction `afficher_nom_prenom` qui prendra en paramètre deux `string` : `nom` et `prenom` qui affichera le nom et le prénom de la personne avec la phrase suivante : « Votre nom est {nom} et votre prénom est {prenom}. ».
- Veuillez créer la fonction `main` qui demandera à l'utilisateur un nom et un prénom, puis les affichera en utilisant la fonction précédente.

```
1 // Quel est votre nom ?
2 // Saisi : Andrew
3 // Quel est votre prénom ?
4 // Saisi : Paul
5 // Votre nom est Andrew et votre prénom est Paul.
```

2 Echange de variables (Pointeurs)

- Créez une fonction `echanger_entiers` qui prendra en paramètres deux pointeurs d'entiers (`int`) et qui inversera les valeurs de ces deux entiers.
- Créez une fonction `afficher_v1_v2` qui prendra en paramètres deux entiers `v1` et `v2`, et qui s'occupera de les afficher à l'écran.
- Réalisez la fonction `main` qui va créer deux entiers avec l'un initialisé à 3 et l'autre à 5. On commencera par les afficher avec la fonction défini ci-dessus. On appellera ensuite la fonction `echanger_entiers` pour échanger ces deux entiers. Enfin, on réaffichera ces entiers pour voir si ils ont bien été inversés.

```
1 // définition : v1=5, v2=3
2 // afficher_v1_v2 : v1=5, v2=3
3 // echange...
4 // afficher_v1_v2 : v1=3, v2=5
```

3 Jeu de tennis (Tableau, enregistrements et pointeurs)

Votre but va être de réaliser le déroulement d'un jeu de Tennis. Comme ce programme est plus dense, un squelette avec des fonctions déjà définies vous est donné à la fin du PDF. Voici les instructions :

- Créez une structure `Joueur` avec trois attributs :
 - « nom » de type `string`.

- « points » de type `int`.
- « gagne_points » de type `bool`.
- Dans la fonction `main`, initialisez un tableau de `Joueurs` au nombre de `NB_JOUEURS`, il s'agit d'une constante définie au début du fichier et qui sera égale à 2.
- Créez la fonction `init_joueurs` qui prendra en paramètre un tableau de `Joueurs`, et qui initialisera les joueurs avec leur nom et mettra à 0 les points, et le drapeau « `gagne_points` » à `false`.
- Appelez ensuite la fonction définie `init_joueurs` dans le `main` afin de pouvoir initialiser les joueurs.
- Créez la fonction `random_point_gagne` qui renverra soit 0, soit 1 pour dire si le point a été gagné ou non.
- Créez la fonction `random_departage_joueur` qui renverra un numéro de joueur de 0 à `NB_JOUEURS-1` et qui sera utilisé si il y'a litige sur le point pour désigner le joueur gagnant du point.
- Créez une fonction `qui_gagne_point` qui prend en paramètre le tableau des joueurs. Cette fonction aura plus effets.
 - Tout d'abord, elle mettra à `true` ou `false` l'attribut « `gagne_points` » de chacun des joueurs pour savoir s'ils ont gagné le point.
 - Ensuite, si plusieurs joueurs ont tous les deux gagnés le point, ou bien qu'aucun d'entre eux ne l'a gagné, on utilisera la fonction `random_departage_joueur` pour départager entre tous les joueurs.
 - Si un seul joueur a gagné le point, il n'y a pas de doute, c'est donc lui qui a gagné le point.
 - La fonction renvoie le numéro du joueur qui a gagné le point.
- Faire une fonction `actualiser_points` qui prendra en paramètre le tableau des joueurs, ainsi que le joueur qui a gagné le point et va modifier son score en conséquence par rapport à son score précédent.
 - 0 => 15
 - 15 => 30
 - 30 => 40
 - 40 => 50 (Fin de la partie)
- Créez la fonction `verifier_gagnant` qui prendra en paramètre le tableau des joueurs et qui vérifiera qu'un des joueurs n'a pas atteint le score de 50. Si un joueur a atteint ce score, il renvoie le numéro du joueur dans le tableau, sinon la fonction renvoie -1.
- Créez la fonction `partie_en_cours` qui prendra en paramètre le tableau des joueurs et qui fera une boucle jusqu'à ce qu'on trouve un gagnant via la fonction `verifier_gagnant`.
 - Cette fonction affichera la score de chacun des joueurs, puis fera dérou-

- ler la partie en appelant `qui_gagne_point`, puis `actualiser_points` jusqu'à qu'on ait un gagnant.
- On retourne le numéro du gagnant.
 - Compléter la fonction `main` pour appeler `partie_en_cours` puis afficher le nom du gagnant de la partie.

```
1  /*
2  Inscription du joueur 1
3  Quel est son nom ? Andrew
4  Inscription du joueur 2
5  Quel est son nom ? Sjha
6  Scores :
7  Andrew : 0 points.
8  Sjha : 0 points.
9  Veuillez appuyer pour continuer.
10 0
11 Scores :
12 Andrew : 0 points.
13 Sjha : 15 points.
14 Veuillez appuyer pour continuer.0
15 Scores :
16 Andrew : 15 points.
17 Sjha : 15 points.
18 Veuillez appuyer pour continuer.0
19 Scores :
20 Andrew : 15 points.
21 Sjha : 30 points.
22 Veuillez appuyer pour continuer.0
23 Scores :
24 Andrew : 30 points.
25 Sjha : 30 points.
26 Veuillez appuyer pour continuer.0
27 Scores :
28 Andrew : 30 points.
29 Sjha : 40 points.
30 Veuillez appuyer pour continuer.0
31 Scores :
32 Andrew : 40 points.
33 Sjha : 40 points.
34 Veuillez appuyer pour continuer.0
35 Le gagnant est : Andrew !
```

4 Carnet des contacts (Tableau, enregistrements et pointeurs)

Votre but va être de construire un carnet de contacts. Vous n'allez pas être guidé cette fois-ci. Cependant, voici ce que le programme devrait faire :

- Demander à l'utilisateur le nombre de contacts qu'il veut dans son carnet. (Utilisez un pointeur de préférence ici.)
- Remplir ensuite les contacts à la chaîne avec un nom et un âge.
- Afficher le carnet d'adresse complet.

Bien sûr, ne faites pas tout ça dans le `main`, utilisez diverses fonctions.

```
1 // Veuillez saisir le nombre de contacts que vous voulez.
2 // Saisi : 3
3
4 // ...Inscription des contacts...
5 // Inscription du contact 1
6 // Quel est son nom ?
7 // Saisi : Paul
8 // Quel est son Âge ?
9 // Saisi : 23
10 // Inscription du contact 2
11 // Quel est son nom ?
12 // Saisi : Andrew
13 // Quel est son Âge ?
14 // Saisi : 26
15 // Inscription du contact 3
16 // Quel est son nom ?
17 // Saisi : Isabelle
18 // Quel est son Âge ?
19 // Saisi : 32
20
21 // ...Affichage des contacts...
22 // Contact n°1 : Paul a 23 ans.
23 // Contact n°2 : Andrew a 26 ans.
24 // Contact n°3 : Isabelle a 32 ans.
```

A Squelette pour l'exercice 3.

```
1 #include <iostream>
2 #include <random>
3 using namespace std;
4
5 // Définir NB_JOUEURS = 2.
6
7 /*
8  * Définir la structure Joueurs.
9  */
10
11 void init_joueurs( ){ // La fonction doit prendre en paramètre
12     // le tableau.
13     for( int i = 0; i < NB_JOUEURS; i++ ){
14         // Initialiser chaque joueur en demandant leur nom, mettre
15         // leur nombre de points = 0 et leur point gagné à false.
16     }
17 }
18
19 int random_point_gagne(){
20
21     random_device dev;
22     mt19937 rng( dev() );
23     uniform_int_distribution <> dist(0,1);
24
25     return dist( rng );
26 }
27
28
29 int random_departage_joueur(){
30
31     random_device dev;
32     mt19937 rng( dev() );
33     uniform_int_distribution <> dist(0,NB_JOUEURS-1);
34
35     return dist( rng );
36 }
```

```
37 }
38
39 int qui_gagne_point( Joueur tab_joueur[] ){
40
41     for( int i = 0; i < NB_JOUEURS; i++ )
42         tab_joueur[i].gagne_points = (bool) random_point_gagne();
43
44     for( int i = 0; i < NB_JOUEURS; i++ ){
45
46         if( i+1 < NB_JOUEURS ){
47             if( tab_joueur[i].gagne_points ==
48                 tab_joueur[i+1].gagne_points )
49                 return random_departage_joueur();
50
51             if ( tab_joueur[i].gagne_points == true )
52                 return i;
53
54         }
55
56         return -1;
57
58     }
59
60     // DÃ©finir verifier_gagnant.
61
62     // DÃ©finir actualiser_points.
63
64     int partie_en_cours( Joueur tab_joueur[] ){
65
66         int gagnant;
67         do {
68
69             int continuer;
70
71             // afficher les scores...
72
73             cout << "Veuillez appuyer sur 0 puis entrÃ©e pour
74                 continuer.";
75             cin >> continuer;
```

```
75 |
76 |     int gagnant = qui_gagne_point( tab_joueur );
77 |     actualiser_points( tab_joueur, gagnant );
78 |
79 | } while( ( gagnant = verifier_gagnant( tab_joueur ) ) == -1 );
80 |
81 |     return gagnant;
82 | }
83 |
84 | int main(){
85 |
86 |     // Faire la fonction main.
87 |
88 |     return 0;
89 | }
```