CPP3 Programmation en C++ - Traitement sur les STRING

Les questions en surbrillance sont à rédiger sur un compte rendu.
Vérifier que la version 17.12 de CodeBlocks est installée, sinon l'installer en exécutant ***codeblocks-17.12mingw-setup.exe*** se trouvant sur le NAS.
La version c++11 (ou supérieur) doit être cochée dans le ***Settings Compiler***

1 – CIN – COUT – Mode DEBUG

⇨ Sous CodeBlock, créer un projet en mode console, langage C++, mode DEBUG et RELEASE. Sauvegarder dans le répertoire de travail.

⇨ Remplacer la fonction main() par la suivante:

```
int main()
{
   int age = 21;
   char pseudo[] = "Mateo";
   cout << "Salut, j'ai " << age << " ans et je m'appelle " << pseudo << endl;
   return 0;
}
```

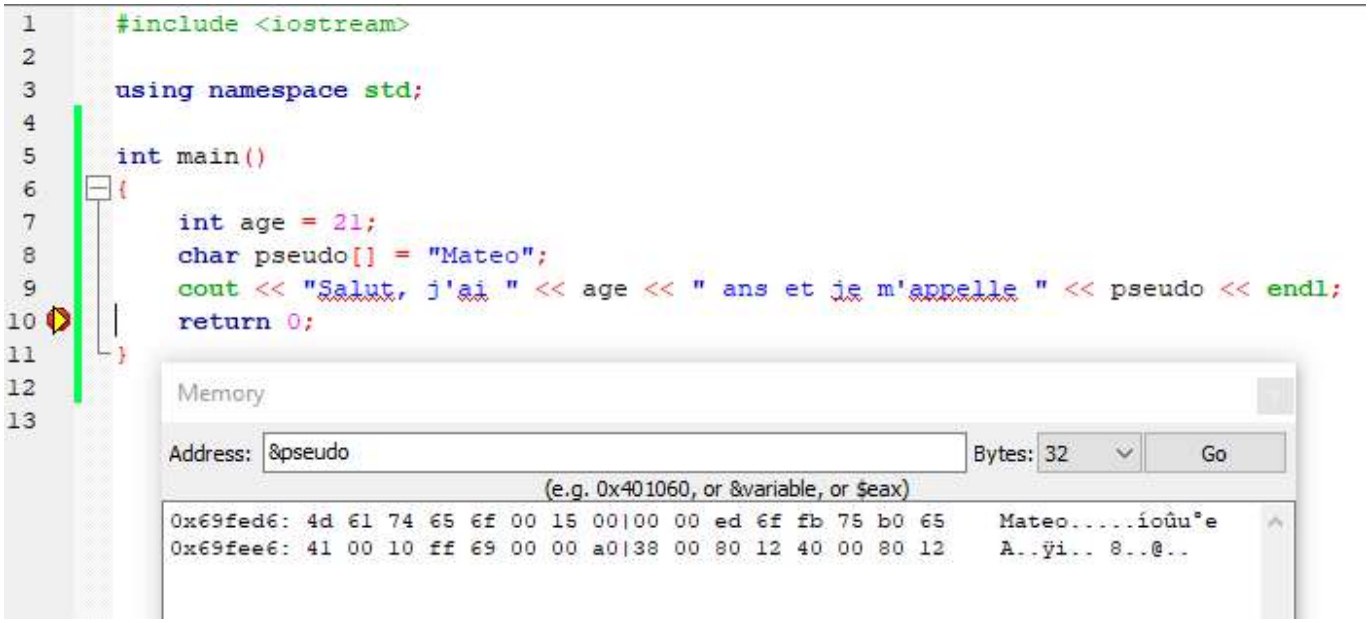
Remarques : cout est la sortie standard sur l'écran pour le C++ ; endl ajoute un saut de ligne.

⇨ Compiler et exécuter le programme.

Remarque : Le compilateur a fait la distinction entre un entier (int) et une chaîne de caractère (char pseudo[]) pour gérer l'affichage des variables.

⇨ Demander une démo au prof pour réaliser ceci :

- Se placer en mode DEBUG
- Placer un point d'arrêt en fin de programme
- Exécuter le programme en mode DEBUG
- Visualiser les variables age et pseudo (Memory Dump)



⇨ Recommencer après avoir changé *int age=21;* par *int age=0x11223344;*

⇨ Combien d'octets occupe la variable age ?

⇨ Indiquer si la chaîne de caractère se termine bien par la valeur 0.

⇨ Ajouter la ligne de programme suivante à la fin (avant return 0;) :

cout << pseudo[2]<<endl;

⇨ Expliquer le résultat affiché (après compilation et exécution).

cin , l'entrée standard en C++, attend des données du clavier.

⇨ Remplacer la fonction main() par la suivante et tester.

```
int main()
{
   int age = 0;
   char pseudo[]="";
   cout << "Comment vous appelez vous ?" << endl;
   cin >> pseudo;
   cout << "Bonjour " <<pseudo<< endl;
   cout << "Quel age avez-vous ?" << endl;
   cin >> age;
   cout << "Ah ! Vous avez " << age << " ans !" << endl;
   return 0;
}
```

Ici aussi le compilateur fait la distinction entre les différents types de données.
La fonction est bloquante puisqu'elle attend des données tapées au clavier.

2 – STRING

Les Strings sont des objets.
On peut donc appliquer des méthodes sur ces objets.
La liste de ces méthodes est ici : http://www.cplusplus.com/reference/string/string/

Exemple : La méthode size() renvoie la longueur de la chaîne de caractères

⇨ Tester l'exemple ci-dessous en remplaçant la fonction main()

```
int main()
{
   string st1;
   cout<<"Entrer une chaine de caracteres: "<<endl;
   cin>>st1;
   cout<<"Nombre de caracteres de la chaine : "<<st1.size()<<endl;
   return 0;
}
```

Initialisation d'un string :

On peut utiliser = pour initialiser un string à une valeur.

Exemples :

```
string st1;
string st2="BONJOUR";
st1="SALUT";
```

Comparaison de 2 strings :

On peut utiliser == pour comparer 2 strings.

Exemples :

```
if (st1==st2) …
```

Exercice :

⇨ Ecrire un programme qui demande un mot de passe à entrer au clavier et qui le compare à un mot de passe initialisé dans le programme. Il doit de plus afficher si le mot de passe entré et correct ou pas.

Concaténation de 2 strings :

On peut utiliser + pour concaténer 2 strings.

⇨ Tester l'exemple ci-dessous :

```
int main()
{
   string st1, st2;
   cout << "Entrer st1:" << endl;
   cin >> st1;
   cout << "Entrer st2:"<< endl;
   cin >> st2;
   cout << "Concatenation de st1 et st2: " << st1+st2 << endl;
   return 0;
}
```

Compatibilité avec les tableaux de données char (chaînes de caractères en C)

On utilise la méthode c_str()

Exemple :

```
int main ()
{
   string st1, st2;
   char c1[]= "BONJOUR";
   const char * c2;
   st1 = c1;
   cout << st1 << endl;
   st2 = "AU REVOIR";
   c2 = st2.c_str();
   cout << c2 << endl;
   return 0;
}
```

st1=c1 : On place le tableau de char c1 dans le string st1

c2=st2.c_str() : On place le contenu du string st2 dans le tableau de char.

3 – Quelques traitements sur les STRING

Conversion d'une chaîne de caractères en entier
Pour convertir une chaîne de caractères en entier, on peut utiliser la fonction stoi()

⇨ Tester l'exemple ci-dessous :

```
#include <iostream>
#include <string>

using namespace std;

int main()
{
   string st1, st2;
   int i1,i2;
   cout << "Entrer st1:" << endl;
   cin >> st1;
   cout << "Entrer st2:"<< endl;
   cin >> st2;
   cout << "Concatenation de st1 et st2: " << st1+st2 << endl;
   i1=stoi(st1);
   i2=stoi(st2);
   cout << "Valeur de i1: "<<i1<<endl;
   cout << "Valeur de i2: "<<i2<<endl;
   cout << "Somme de i1 et i2: " << i1+i2 << endl;
   return 0;
}
```

⇨ A l'aide des informations recueillies sur le site cplusplus.com, modifier les instructions stoi du programme ci-dessus pour pouvoir convertir en entier des chaînes de caractères représentant des nombres hexadécimaux. Tester.

Trouver un caractère dans une chaîne
On utilise la méthode find(). La valeur retournée correspond à la position du caractère recherché.

⇨ Tester l'exemple ci-dessous:

```
#include <iostream>
#include <string>

using namespace std;

int main()
{
   string st1;
   st1="CODE=1234";
   size_t pos=st1.find("=");
   cout<<"position de la recherche: "<<pos<<endl;
   return 0;
}
```

⇨ Justifier la valeur affichée

⇨ Modifier le caractère recherché "=" par le caractère "3" et justifier la valeur affichée.

⇨ Modifier la chaîne par la valeur "CODE=1234563898" et faire une recherche sur le caractère "3".

⇨ Conclure sur le résultat obtenu.

Une recherche peut se faire sur plusieurs caractères.
⇨ Avec la même chaîne que précédemment, faire une recherche sur "38"
⇨ Conclure

Une recherche peut se faire à partir d'une position
⇨ Avec la chaîne "CODE=1234563898", faire une recherche sur le caractère 3 à partir de la position 8 : st1.find("3",8).
⇨ Expliquer le résultat obtenu.

Extraire une partie d'une chaîne de caractère
On utilise la méthode substr(position du premier caractère, longueur de la chaîne à extraire)

⇨ Tester l'exemple ci-dessous :

```
#include <iostream>
#include <string>

using namespace std;

int main()
{
   string st1, st2;
   st1="CODE=123456";
   st2=st1.substr(5,6);
   cout<<"Chaine extraite : "<<st2<<endl;
   return 0;
}
```

⇨ Expliquer le résultat obtenu.

⇨ Modifier st1 avec la valeur "CODE SECRET=123456"

⇨ Remplacer l'instruction `st2=st1.substr(5,6);` par l'instruction `st2=st1.substr(5);` (sans le 2ème argument).
⇨ Conclure sur le résultat obtenu.

Exercice :
Proposer un programme qui extrait d'une chaîne st1, deux chaînes : une contenant les caractères avant le signe "=" et l'autre contenant les caractères après le signe "=".
Exemple : si st1="CODE_MAISON=12458", la chaîne sera divisée en 2 chaînes : "CODE_MAISON" et "12458".
Il faudra, pour cela, utiliser la méthode substr() vue précédemment pour déterminer la position du signe "=".

That's all folks