

TP 1 TP d'initiation

Connectez-vous sous Linux.

Les cours, TD et TP sont accessibles sur l'ENT dans le cours **R1.01 - Initiation au développement**

Ouvrez une fenêtre Terminal (Applications → Utilitaires), puis placez vous sous le répertoire Documents (`cd Documents`).

Ensuite créez un répertoire algo (`mkdir algo`).

Positionnez vous sur ce répertoire (`cd algo`).

Créez un nouveau répertoire TP1 (`mkdir TP1`), puis positionnez vous sur ce répertoire. Vous créez un répertoire de TP pour chaque séance.

Ouvrez l'éditeur de texte VS code.

Déplacez-vous dans le répertoire de travail : File => Open Folder => navigation ...

1 - Premier Programme

Soit le programme source suivant, écrit en langage C :

```
/**
 * @brief mon premier programme C
 */
#include <stdio.h>
#include <stdlib.h>
int main() // debut du programme
{
    printf("Bonjour !!\n");
    return EXIT_SUCCESS;
} // fin du programme
```

1. Saisissez ce programme dans un fichier source et sauvegardez-le en le nommant **hello.c**.
2. Compilez ce programme par la commande ci-dessous tapée dans la fenêtre "terminal" :
Terminal => New Terminal
`cc hello.c -o hello -Wall`
3. Exécutez le programme par la commande ci-dessous tapée dans la fenêtre "terminal" :

```
./hello
```

Observez les fichiers créés en tapant `ls` dans le terminal. Observez le fichier exécutable `hello` en l'ouvrant dans un éditeur de texte.

2 - Lecture au clavier et écriture à l'écran

Soit le programme source suivant, écrit en langage C :

```
/**
 * @brief pg qui demande un nom et un age et qui les affiche
 */
#include <stdio.h>
#include <stdlib.h>

int main()
{
    // Declaration des variables
    char nom[20]; // une chaîne d'au plus 20 caractères (dont le caractère de fin
de chaîne, donc 19 disponibles)
    int age;      // un entier
    // Saisie des donnees
    printf("Bienvenue au département informatique\n");
    printf("Quel est votre nom (au plus 19 caractères, et sans espace) ? : ");
    scanf("%s", nom);
    printf("quel age as tu ?\n");
    scanf("%d", &age);
    // Affichage
    printf("Bonjour %s\n", nom);
    printf("Tu as %d année(s)\n", age);
    return EXIT_SUCCESS;
}
```

Après avoir saisi ce programme, nommez le fichier puis compilez et exécutez ce programme et observez son exécution.

Remarques :

- les instructions s'exécutent en séquence.
- il est possible d'afficher des expressions (par exemple une différence d'âge).

Testez et comprenez le programme suivant :

```
/**
 * @brief deux ages sont demandés et leur différence affichée
 */
#include <stdio.h>
#include <stdlib.h>
int main(){
    // Declaration des variables
    char nom1[50]; // nom d'une première personne (50 car, dont le caractère de fin de
chaîne)
    int age1;      // déclaration de l'age de la personne numero 1
    char nom2[50]; // nom d'une deuxième personne
    int age2;      // déclaration de l'age de la personne numero 2
    // Saisie des donnees
    printf("Bienvenue au département informatique\n");
    printf("Personne numero 1, votre nom SVP (au plus 49 car, et sans espace) : ");
    scanf("%s", nom1);
    printf("Personne numero 1, votre age SVP : ");
    scanf("%d", &age1);
    printf("Personne numero 2, votre nom SVP (au plus 49 car, et sans espace) :");
    scanf("%s", nom2);
    printf("Personne numero 2, votre age SVP : ");
    scanf("%d", &age2);
    // Affichage du resultat
    printf("Vous avez %d annees de difference.\n", age2 - age1);
    return EXIT_SUCCESS;
}
```

Si vous tapez plus de 50 caractères dans les chaînes de caractères, vous provoquerez un débordement et risquez de faire planter le programme. Essayez avec différentes quantités de caractères.

3 - Affectation, expressions, variables et constantes

a- Affectation et séquentialité

Le but de cette partie est de vérifier que vous avez bien compris que les instructions (les commandes) que vous donnez à l'ordinateur par l'intermédiaire du langage C sont exécutées dans l'ordre séquentiel (de haut en bas) dans lequel vous rédigez ces instructions.

Nous allons d'abord commencer par l'affectation et ses contraintes au travers du programme suivant (à tester) :

```
/**
 * @brief quelques affectations
 */
#include <stdio.h>
#include <stdlib.h>
// Declaration des constantes
const float PI=3.14159; // déclaration et affectation d'une constante de type réel
int main(){
    // Declaration des variables
    int un ; // déclaration d'une variable de type entier
    float x; // déclaration d'une variable de type réel
    // Initialisation des donnees
    un = 12;
    x = 8.42;
    // Affichage
    printf("voici un : %d et x : %f. Ce sont deux variables.\n", un,x); // quelle
différence avec la suite ?
    printf("voici un : %d", un);
    printf(" et x : %f. Ce sont deux variables.", x);
    printf("\n");
    printf("voici un : %.3f", PI);
    return EXIT_SUCCESS;
}
```

Quelle est la différence entre les deux affichages ? Pouvez-vous proposer d'autres écritures qui aboutissent au même affichage ?

b - Une recette de gâteau

Le programme suivant vous affichera la quantité d'ingrédients nécessaire pour la réalisation d'un far breton pour 8 personnes

```
/**
 * @brief Recette de far breton pour 8 personnes
 */
#include <stdio.h>
#include <stdlib.h>
int main(){
    // variables contenant les quantités
    float pruneaux, farine, sucre, lait;
    int oeufs;
    // initialisation des quantités
    pruneaux = 250.0;
    farine = 200.0;
    sucre = 150.0;
    oeufs = 4;
    lait = 0.75;
    // affichage des quantités
    printf("Pour préparer un Far Breton pour 8 personnes, il faut :\n");
    printf(" %f grammes de pruneaux\n", pruneaux);
    printf(" %f grammes de farine\n", farine);
    printf(" %f grammes de sucre\n", sucre);
    printf(" %d oeufs", oeufs);
    printf(" %f litre(s) de lait", lait);
    return EXIT_SUCCESS;
}
```

Tapez ce programme, compilez le et observez le résultat.

Pour avoir une meilleure présentation des nombres réels, mettez `%5.1f` au lieu de `%f` : sur 5 caractères dont 1 chiffre après le point décimal.

c- Une recette de gâteau adaptable au nombre de convives

Nous voudrions maintenant que le nombre de personnes pour lesquelles est fait le gâteau puisse influencer sur la quantité d'ingrédients. Pour cela nous allons indiquer, par l'intermédiaire d'une valeur demandée à l'utilisateur, le nombre de personnes.

Nous allons donc demander le nombre de personnes à l'utilisateur grâce à l'instruction `scanf`, que nous allons stocker dans une **variable** nommée `nbpers` et réutiliser cette variable pour faire des calculs. Plus exactement, il suffira de multiplier les quantités précédente par le nombre de personnes divisé par 8.

Modifiez votre programme en conséquence.

4 – Cas particuliers des lectures clavier

a – saisie trop longue

Tapez, compilez puis exécutez ce programme en saisissant une chaîne de 15 caractères. Que se passe-t-il ? Pourquoi ?

```
int main()
{
    // Declaration des variables
    char s1[10] = "robert";
    char s2[10];
    // Saisie des donnees
    printf("S2 ? ");
    scanf("%s", s2);
    // Affichage
    printf("S1=%s\n", s1);
    printf("S2=%s\n", s2);

    return EXIT_SUCCESS;
}
```

b – caractères séparateurs

Tapez, compilez puis exécutez ce programme en saisissant "DE NIRO" pour la première chaîne et "ROBERT" pour la deuxième chaîne. Que se passe-t-il ? Pourquoi ?

```
int main()
{
    // Declaration des variables
    char nom[10];
    char prenom[10];
    // Saisie des donnees
    printf("Nom ? ");
    scanf("%s", nom);

    printf("Prenom ? ");
    scanf("%s", prenom);

    // Affichage
    printf("Nom=%s\n", nom);
    printf("Prenom=%s\n", prenom);

    return EXIT_SUCCESS;
}
```

c – le "retour chariot"

Tapez, compilez puis exécutez ce programme. Que se passe-t-il ? Pourquoi ?

```
int main()
{
    // Declaration des variables
    char nom[10];
    char groupe;

    // Saisie des donnees
    printf("Nom ? ");
    scanf("%s", nom);

    printf("Groupe ? ");
    scanf("%c", &groupe);
    // Affichage
    printf("Nom=%s\n", nom);
    printf("Groupe=%c\n", groupe);

    return EXIT_SUCCESS;
}
```

5 - Exercices complémentaires

a – montant TTC

Écrivez et testez un programme qui demande à l'utilisateur un prix hors taxe, un taux de taxe (en pour-cent) puis qui affiche le montant TTC à l'écran.

b – montant hors taxe

Écrivez et testez un programme qui calcule et affiche un prix hors taxe, le montant de la taxe à partir du montant TTC et du taux de la taxe (en pour-cents) qui seront saisis au clavier par l'utilisateur.

c – affichage d'une différence en pourcentage

Sauriez-vous écrire un programme auquel on fournit deux prix (pour le même produit, par exemple dans deux magasins différents ou à des dates différentes) et qui indique la différence de prix en %. Par exemple, on saisit 100 et 125, et le programme affiche qu'il y a 25% de hausse. Testez-le avec des valeurs différentes et y compris des valeurs qui le font échouer.

d- Écrivez et testez un programme qui affiche le dessin suivant fait avec des lettres :

```
  A
  A  A
  A    A
AAAAAAA
```

L'utilisateur saisira le caractère à afficher.

e- Moyenne des températures

Écrivez et testez un programme qui demande quatre réels qui correspondent à quatre prises de température dans la journée (à 8:00, à 10:00, à 14:00, à 16:00). On affichera la moyenne des deux températures du matin, la moyenne des deux températures de l'après-midi et la moyenne de toutes les températures.

f- Quel âge en 2030 ?

Écrivez et testez un programme qui demande à l'utilisateur son année de naissance et affiche l'âge qu'il aura lors de son anniversaire de 2030.