

TP 2

Adresse du cours : <https://axlbonnet.github.io/dut-gim-pres>

TP2-1

Ecrire un algorithme pour obtenir un montant en euros, et afficher sa conversion en francs (1€ = 6.55956 francs). On ne souciera pas du nombre de chiffres après la virgule.

TP2-2

Ecrire un algorithme pour calculer le valeur absolue de “a – b”. a et b sont des nombres réels et doivent être saisis par l'utilisateur.

TP2-3

Ecrire un algorithme pour déterminer si l'utilisateur est du signe astrologique balance (du 23 septembre inclu au 23 octobre exclu).

Programmes en C

Rappel : sauvegardez tous vos programmes dans le lecteur réseau“W:”

On considérera que toutes les saisies de l'utilisateurs sont bonnes.

TP2-4

Utilisez le méthode “printf” pour écrire le texte suivant (dièses non compris). Les nombres “89”, “1,99”, “67” et “1” seront saisis par l'utilisateur. Utiliser dans un 1er temps 3 appels à “printf” (un par logne), un seul appel dans un second temps.

```
#####
```

```
Bonjour toi.
```

```
Ceci est un texte bidon avec les chiffres 42, 89 et 1.990000.
```

```
Je suis sur a 67% que ca ne veut rien dire, ca vaut 1/20.
```

```
#####
```

TP2-5

Appliquer les 3 premiers exercices (TP2-1 P2-2 TP2-3) en C.

TP2-6

Ecrire un programme pour obtenir un montant entier en euros, et afficher sa conversion en différentes devises (1€ = 6.55956 francs / 1,19583\$ / 0,900614£). Transformer ensuite le programme pour que l'utilisateur choisisse sa devise (avec un nombre saisi au clavier, par exemple 1 pour franc, etc).

TP2-7

Ecrire un programme pour demander le jour et le mois de naissance, puis la taille de l'utilisateur.

Afficher le texte suivant “Bonjour ! Tu es ne en automne et tu mesure 1.80000m” en adaptant la saison.

TP 2 (2eme partie)

Algorithmes

TP2-8

Ecrire un algorithme pour obtenir 5 fois les dimensions d'un triangle (base et hauteur). Afficher à chaque fois l'aire du triangle.

TP2-9

Ecrire le même algorithme que le précédent, sauf que le nombre d'aires de triangle à afficher est obtenu auprès de l'utilisateur.

TP2-10

Ecrire un algorithme qui demande un nombre de notes à saisir, invite l'utilisateur à les saisir, puis calcule et affiche leur moyenne.

TP2-11

Améliorer l'exercice précédent en faisant en sorte que les notes inférieures à 10 ont un coefficient 3.

TP2-12

Ecrire un algorithme qui demande un nombre entier à l'utilisateur, puis vérifie si ce nombre est le multiple d'un nombre compris entre 1 et 101. Les divisions et les modulus sont interdits, les multiplications sont les seules opérations mathématiques autorisées.

Programmes en C

On considérera que toutes les saisies de l'utilisateurs sont bonnes.

TP2-13

Appliquer les exercices précédents en C.

TP2-14

Afficher la table de multiplication des nombres entre 1 et 13. Vous pourrez tester et utiliser le caractère de la tabulation (“\t”) pour aligner les colonnes.

TP2-Bonus

Programmez un super-morpion. 2 joueurs entrent chacun leur tour un nombre entre 1 et 9 sur une grille 3x3. Le premier joueur utilise uniquement des nombres impairs, le deuxième uniquement des nombres pairs. Chaque nombre ne peut être utilisé qu'une seule fois. Le premier joueur qui obtient exactement 15 points sur une ligne (ou diagonale) gagne.