

**TP 2 - PROGRAMMATION MODULAIRE**Rappels généraux :

- ↳ Un programme non commenté et dont l'indentation n'est pas correcte ne sera pas lu.
- ↳ Cocher l'option –Wall pour afficher tous les avertissements du compilateur.
- ↳ Pour ce TP, vous pourrez vous référer au chapitre VII du Manuel de Survie.
- ↳ Récupérer le fichier compressé contenant les débuts de programme sur Moodle. Décompressez ce fichier dans un nouveau dossier pour pouvoir ensuite modifier les programmes. A la fin de la séance, vous penserez à placer ce dossier ainsi que le fichier compressé téléchargé dans la corbeille, puis à la vider

**1. PGCD selon Euclide****niv 1**

Ouvrir le fichier tp2exo1.c. Compléter ce programme en y ajoutant la déclaration du prototype de la fonction pgcd puis le programme principal dans lequel on demande à l'utilisateur de rentrer un nombre entier, on calcule et affiche le pgcd de ce nombre et 12, on demande ensuite un 2<sup>nd</sup> nombre entier à l'utilisateur, on calcule et affiche alors le pgcd du 1<sup>er</sup> et du 2<sup>nd</sup> puis de 3 fois le 1<sup>er</sup> et 12 fois le 2<sup>nd</sup>.

```

1  //-----
2  //TP 2 Exercice 1
3  #include <stdio.h>
4
5  //prototypes des fonctions
6  int pgcd(int,int);
7
8
9  //programme principal
10 int main(void)
11 {
12     int a,b,p;
13
14     printf("entrez un nombre entier : ");
15     scanf("%d",&a);
16
17     p=pgcd(a,12);
18     printf("le pgcd de %d et %d est %d\n",a,12,p);
19
20     printf("entrez un nombre entier : ");
21     scanf("%d",&b);
22
23     p=pgcd(a,b);
24     printf("le pgcd de %d et %d est %d\n",a,b,p);
25     printf("le pgcd de %d et %d est %d\n",3*a,12*b,pgcd(3*a,12*b));
26
27     return 0;
28 }
29
30 //fonction renvoyant le pgcd de ses 2 entrees
31 int pgcd(int a,int b)
32 {
33     while (a*b) {
34         if (a>b)
35             a-=b;
36         else
37             b-=a;
38     }
39
40     return (a+b);
41 }
```

**2. Fonction Altitude****niv 1**

1. Ouvrir le fichier tp2exo2.c. Compléter ce programme en y ajoutant la définition de la fonction altitude, qui renvoie l'altitude d'un objet en fonction du temps  $t$  et de sa vitesse initiale  $v_0$  selon la loi :

$$z(t) = \frac{-g}{2}t^2 + v_0t$$

2. Modifier ensuite le programme de façon à ce que toutes les variables soient déclarées comme réelles

```

1  //-----
2  //TP 2 Exercice 2
3  #include <stdio.h>
4  #define g 9.81 //acceleration de la pesanteur en m.s-2
5
6  //prototypes des fonctions
7  double altitude(double, double);
8  void accueil(void);
9
10 //programme principal
11 int main(void)
12 {
13     double z, t, v0;
14
15     accueil();
16     printf("entrez la vitesse initiale (en m.s-1) : ");
17     scanf("%lf", &v0);
18     printf("entrez le temps (en s) : ");
19     scanf("%lf", &t);
20
21     z = altitude(v0, t);
22     printf("au bout de %.2f s, l'objet sera a une altitude de %.2f m\n", t, z);
23     //le format %.2f permet de n'afficher que 2 chiffres apres la virgule
24     return 0;
25 }
26
27 //fonction affichant le message d'accueil
28 void accueil(void)
29 {
30     printf("#####\n");
31     printf("#                               #\n");
32     printf("#   Altitude d'un objet en chute libre   #\n");
33     printf("#   avec vitesse initiale verticale   #\n");
34     printf("#                               #\n");
35     printf("#####\n\n");
36 }
37
38 //fonction calculant l'altitude à un instant t donne
39 double altitude(double v, double t)
40 {
41     return (-g/2.*t*t+v*t);
42 }

```

### 3. Conversions de température

**niv 1**

La température Celsius(centigrade) peut être convertie en Fahrenheit par la formule suivante :

$$F = 9/5 C + 32$$

1. Ecrire le code de la fonction Cels2Fahr qui convertit une température Celsius en Fahrenheit.
2. Ecrire le code de la fonction Fahr2Cels qui effectue l'opération inverse.
3. Ecrire le programme principal dans lequel on demande à l'utilisateur de saisir une température en degrés Celsius, on la convertit et on l'affiche en degrés Fahrenheit puis on la reconvertit en degrés Celsius.

```

1  //-----
2  //TP 2 Exercice 3
3  #include <stdio.h>
4
5  //prototypes des fonctions
6  double Cels2Fahr(double);
7  double Fahr2Cels(double);
8
9  //programme principal
10 int main(void) {
11     double TCels, TFahr;
12
13     //entree de la température en Celsius
14     printf("Veuillez entrer la température en °C : ");
15     scanf("%lf",&TCels);
16
17     //conversion de la température en Fahrenheit
18     TFahr=Cels2Fahr(TCels);
19     printf("voici la température convertie en Fahrenheit : %.1f°F\n",TFahr);
20
21     //reconversion de la temperature en Celsius
22     TCels=Fahr2Cels(TFahr);
23     printf("voici la température reconvertie en Celsius : %.1f°C\n",TCels);
24
25     return 0;
26 }
27
28 //fonction Cels2Fahr convertissant une température en °C en °F
29 double Cels2Fahr(double T)
30 {
31     double T2;
32     T2=9./5.*T+32.;
33     return T2;
34 }
35
36 //fonction Fahr2Cels convertissant une température en °F en °C
37 double Fahr2Cels(double T)
38 {
39     return 5./9.*T-160/9.;
40 }

```

#### 4. Fonctions Carré et puissance4

**niv 2**

Ecrire :

1. Une fonction `carre` qui calcule le carré d'un nombre entier donné.
2. Une fonction `puissance4` qui élève un nombre entier passé en argument à la puissance 4. Votre fonction ne devra contenir qu'une seule fois le signe `*` (voire aucune fois).
3. Le programme principal qui demande un nombre entier à l'utilisateur et affiche son carré et ce nombre à la puissance 4.

Votre programme ne devra faire appel à aucune fonction de la bibliothèque mathématique.

```
1 //-----
2 //TP 2 Exercice 4
3 #include <stdio.h>
4
5 int carre(int);
6 int puissance4(int);
7
8 int main(void)
9 {
10     int a,a2,a4;
11
12     printf("entrez une valeur entiere : ");
13     scanf("%d",&a);
14
15     a2=carre(a);
16     a4=puissance4(a);
17
18     printf("a puissance 2 = %d\na puissance 4 = %d\n",a2,a4);
19
20     return 0;
21 }
22
23 int carre(int x)
24 {
25     return x*x;
26 }
27
28 int puissance4(int x)
29 {
30     return carre(x)*carre(x);
31     return carre(carre(x)); //2ème solution
32     return carre(x*x); //3ème solution
33 }
```

## 5. Fonctions aléatoires

**niv 2**

1. Ecrire un programme principal qui affiche la valeur de la constante `RAND_MAX` (cette valeur est définie dans la bibliothèque `stdlib`, il est donc illicite de la redéfinir).
2. Ecrire une fonction `dice6` qui renvoie un nombre entier au hasard entre 1 et 6. La tester dans le programme principal.
3. Ecrire une fonction `edice` qui renvoie un nombre entier au hasard entre 1 et une valeur maximale donnée en paramètre de la fonction. La tester dans le programme principal.
4. Ecrire une fonction `aleat` qui renvoie un entier au hasard entre 2 valeurs passées comme argument (on admettra que les 2 nombres en entrée sont rangés dans l'ordre croissant). La tester dans le programme principal.
5. Ecrire une fonction qui renvoie une lettre minuscule au hasard. La tester dans le programme principal.

### Annexe : nombres aléatoires

Pour générer aléatoirement une suite  $X_i$  de nombres aléatoires, on utilisera la fonction `rand()` définie dans la bibliothèque mathématique qui renvoie un nombre entier au hasard entre 0 et `RAND_MAX` (défini dans `stdlib.h`). Il est également nécessaire d'appeler `srand(time(0))` au moins une fois avant l'utilisation de `rand()` (la fonction `time` est définie dans la bibliothèque `time.h`).

*Idées :*

- ↳ si  $n$  est un entier tiré au hasard entre 0 et `RAND_MAX`, alors  $n\%p$  est un entier tiré au hasard entre 0 et  $(p-1)$  (en supposant  $p \ll \text{RAND\_MAX}$ )
- ↳ si  $n$  est un entier tiré au hasard entre 0 et `RAND_MAX`, alors  $(\text{float}) n / (\text{RAND\_MAX} + 1.0)$  est un nombre flottant compris dans l'intervalle  $[0, 1[$

```

1  //-----
2  //TP2 Exercice 5
3  #include <stdio.h>
4  #include <time.h>
5  #include <stdlib.h>
6
7  //prototypes des fonctions
8  int dice6(void);
9  int edice(int);
10 int aleat(int,int);
11 char letminrand(void);
12
13 //programme principal
14 int main(void)
15 {
16     int val,min,max,n;
17
18     srand(time(0)); //initialisation des nombres aleatoires
19
20     printf("valeur max des nombres aleatoires : %d\n\n",RAND_MAX);
21
22     val=dice6();
23     printf("nombre aleatoire entre 1 et 6 : %d\n",val);
24
25     printf("entrez le nombre de faces du de : ");
26     scanf("%d",&n);
27     val=edice(n);
28     printf("nombre aleatoire entre 1 et %d : %d\n",n,val);
29
30     printf("entrez une valeur min puis une valeur max : ");
31     scanf("%d%d",&min,&max);
32     val=aleat(min,max);
33     printf("nombre aleatoire entre %d et %d : %d\n",min,max,val);
34
35     printf("lettre minuscule aleatoire : %c\n",letminrand());
36     return 0;
37 }
```

```
38
39 //definition des fonctions
40 int dice6(void) {
41     return (1+rand()%6);
42 }
43
44 int edice(int a)
45 {
46     return (1+rand()%a);
47 }
48
49 int aleat(int a,int b)
50 {
51     int r;
52     r=rand(); //r est compris entre 0 et RANDMAX
53     r=r%(b-a+1); //r est maintenant compris entre 0 et b-a;
54     r=r+a; //r est maintenant compris entre a et b
55     return r;
56 }
57
58
59 char letminrand(void)
60 {
61     return (char) ('a'+rand()%26); //on part de 'a' et il y a 26 lettres
62     return (char) aleat('a','z'); //autre solution
63 }
```

## 6. Temps de vol

**niv 3**

1. Ecrire une fonction qui calcule la durée de vol en minutes connaissant l'heure de départ et l'heure d'arrivée (sous la forme heure et minutes).
2. Ecrire une fonction qui affiche le temps de vol en heures et minutes connaissant le temps de vol en minutes.
3. Ecrire un programme qui demande les heures de départ et d'arrivée à l'utilisateur et affiche le temps de vol.
4. Reprendre la 1<sup>ère</sup> fonction en supposant maintenant que la durée de vol est inférieure à 24 heures mais que l'arrivée peut avoir lieu le lendemain.

```

1 //-----
2 //TP2 Exercice 6
3 #include <stdio.h>
4
5 int cal_temps_vol(int hd,int md,int ha,int ma);
6 void aff_temps_vol(int);
7
8 int main(void)
9 {
10     int hd,md,ha,ma,tvol;
11
12     printf("entrez l'heure du depart : ");
13     scanf("%d",&hd);
14     printf("entrez minute du depart : ");
15     scanf("%d",&md);
16     printf("entrez l'heure d'arrivee : ");
17     scanf("%d",&ha);
18     printf("entrez minute du d'arrivee : ");
19     scanf("%d",&ma);
20
21     tvol=cal_temps_vol(hd,md,ha,ma);
22     printf("\nduree du vol : ");
23     aff_temps_vol(tvol);
24
25     return 0;
26 }
27
28 int cal_temps_vol(int hd,int md,int ha,int ma)
29 {
30     int ta,td;
31     //on ajoute 24 heures si l'arrivee est avant le depart
32     if (ha<hd || (ha==hd && ma<=md))
33     {
34         ha=ha+24;
35     }
36
37     td=60*hd+md; // instant du depart en minutes
38     ta=60*ha+ma; // instant d'arrivee en minutes
39     return (ta-td);
40 }
41
42 void aff_temps_vol(int t)
43 {
44     int h,m;
45     h=t/60;
46     m=t%60;
47     printf(" %02dh%02d ",h,m);
48 }

```