

TP 1: Algorithmique procédurale

Concepts de base

Nga Nguyen - Stefan Bornhofen - Peio Loubière

1 Concepts de base

L'ensemble des sorties écran doivent être explicites (afficher une phrase formatée).

Question 1 : Écrire un programme qui affiche à l'écran "Bonjour le monde!\n"

```
1 int main(int argc, char** argv){
2     //affichage à l'écran
3     printf("Bonjour le monde!");
4 }
```

Question 2 : Étudiez, compilez et exécutez le code suivant; que remarquez-vous?

```
1 #include <stdio.h>
2 int main (int argc, char** argv) {
3     int int_entier; //Variable de test
4
5     printf ("Valeur de l'entier : %d\n",
6             int_entier);
7
8     // Retourne 0 car le programme ne plante pas
9     return (0);
}
```

affiche un entier. L'entier n'est pas initialisé mais pourtant il possède une valeur.

Question 3 : Écrire un programme qui demande la saisie de 2 nombres et calcule la moyenne

```

1 int main (int argc, char** argv) {
2     float float_note1; // La premiere note
3     float float_note2; // La seconde note
4
5     // Affichage de ce que l'on demande
6     printf ("Quel est la premiere note: ");
7     // Demande de la premiere note à l'utilisateur
8     scanf("%f", &float_note1);
9     // Affichage de ce que l'on demande
10    printf ("Quel est la seconde note: ");
11    // Demande de la seconde note à l'utilisateur
12    scanf("%f", &float_note2);
13
14    printf ("La moyenne de %f et %f est %f\n",
15           float_note1, float_note2, ((float_note1 +
16           float_note2) / 2));
17
18    // Retourne 0 car le programme ne plante pas
19    return (0);
20 }

```

Question 4 : Écrire un programme qui calcule et affiche la valeur de l'hypothénuse d'un triangle rectangle, connaissant 2 de ses côtés

```

1 int main (int argc, char** argv) {
2     float float_cote1; // Le premier cote
3     float float_cote2; // Le second cote
4
5     // Affichage de ce que l'on demande
6     printf ("Quel est la longueur du premier cote : \n"
7             );
8     // Demande du premier cote à l'utilisateur
9     scanf("%f", &float_cote1);
10    // Affichage de ce que l'on demande
11    printf ("Quel est le longueur du second cote : \n");
12    // Demande du seconde cote à l'utilisateur
13    scanf("%f", &float_cote2);
14
15    // Affichage du message
16    printf ("L'hypothénuse vaut %f\n", sqrtf ( (
17            float_cote1 * float_cote1) + (float_cote2 *
18            float_cote2)));
19
20    // Retourne 0 car le programme ne plante pas
21    return (0);
22 }

```

Question 5 : Écrire un programme qui demande l'heure et qui doit répondre "Dans une minute, il sera...". (Vérifiez la validité de l'heure avant d'effectuer le traitement)

```

1 int main (int argc, char** argv) {
2     int h,m; // Le temps a saisir
3
4     // Affichage de ce que l'on demande
5     printf (" Saisir les heures : ");
6     scanf ("%d", &h);
7     printf (" Saisir les minutes : ");
8     scanf ("%d", &m);
9
10    m = m + 1;
11    if (m == 60) {
12        m = 0;
13        h ++;
14    }
15    if (h == 24)
16        h = 0;
17    printf (" Dans une minute il sera %d heure (s) %d
18           minute (s)", h,m);
19 }

```

Question 6 : Écrire un programme qui permet de saisir une série de nombres jusqu'à que l'utilisateur saisisse "0". Il affiche ensuite le plus grand nombre saisi

```

1 int main (int argc, char** argv) {
2     int int_max, int_nb; // nb saisi
3     int_max = -10000; // initialisation du max
4
5     do{
6         // Affichage de ce que l'on demande
7         printf ("Saisir un nombre: ");
8         // Demande du premier cote à l'utilisateur
9         scanf("%d", &int_nb);
10        //si le nb est plus grand que le max, le max
11        //devient le nombre
12        if (int_nb > int_max)
13            int_max = int_nb;
14    }while(int_nb != 0) //tant que le nombre saisi
15    //est différent de zero
16
17    // Affichage du message
18    printf ("Le plus grand nombre saisi est %d\n",
19            int_max);
20
21    // Retourne 0 car le programme ne plante pas
22    return (0);
23 }

```

Question 7 : Écrire un programme qui demande la saisie de 2 nombres entiers et réalise la division entière. Ce programme doit détecter et afficher un message :

- si l'utilisateur essaie de diviser par zéro
- si l'utilisateur ne saisit pas un entier

Tester votre programme avec les valeurs (4,2), (7,3), (8,0), (12.6,2) et (7,3.5)

```

1 int main (int argc, char** argv) {
2     int nb1, nb2, retour;
3
4     // Affichage de ce que l'on demande
5     do{
6         printf ("Quel est le premier nombre: ");
7         retour=scanf("%d", &nb1);
8     }while(retour!=1)
9     do{
10    // Affichage de ce que l'on demande
11    printf ("Quel est le second nombre: ");
12    scanf("%d", &nb2);
13    }while(retour!=1)
14
15    if(nb2 !=0)
16        printf ("%d/%d est %d\n", nb1, nb2, nb1/nb2));
17    else
18        printf ("Division par zéro!");
19
20    // Retourne 0 car le programme ne plante pas
21    return (0);
22 }

```

2 Fonctions-Boucles

Question 8 : Écrire la fonction **table** permettant d'afficher la table de multiplication d'un nombre passé en paramètre

```

1 void table(int nb){
2     int i;
3     printf ("Table de %d", nb);
4     for (i=1; i ≤ 10; i++){
5         printf ("%d*%d = %d\n", nb, i, i*nb);
6     }
7 }

```

Question 9 : Écrire la fonction **factorielle** calculant la factorielle d'un nombre

```
1 int factorielle(int nb){
2     int i, res;
3     res = 1;
4     for (i=1; i ≤ nb; i++){
5         res *= i;
6     }
7     return res;
8 }
```

Question 10 : Écrire la fonction **parfait** qui teste si un nombre est parfait ou non

```
1 int parfait(int nb){
2     int i, res;
3     res = 0;
4     for (i=1; i ≤ nb; i++){
5         if (nb % i != 0)
6             res += i;
7     }
8     return (res==nb);
9 }
```

Question 11 : Écrire le programme permettant de tester ces fonctions, il propose un menu avec choix, chacun appelant la fonction souhaitée, le dernier permettant de quitter.

Vous devez contrôler la saisie et reposer le menu si le choix saisi n'est pas dans l'intervalle proposé ou n'est pas un entier

```

1  int main (int argc, char** argv) {
2      int rep;
3
4      // Affichage de ce que l'on demande
5      do{
6          printf ("1.: Table.");
7          printf ("2.: Factorielle.");
8          printf ("3.: Parfait.");
9          printf ("Autre.: Quitter.");
10         printf ("Quel est votre choix:");
11         retour=scanf("%d", &rep);
12         if(retour==1){
13             printf ("Saisir un nombre.");
14             retour=scanf("%d", &nb);
15             switch(rep){
16                 case 1: table(nb);
17                     break;
18                 case 2: printf("%d!=", factorielle(nb));
19                     break;
20                 case 3: if (parfait (nb)!=0)
21                         printf("%d est un nombre
22                             parfait",nb);
23                     else
24                         printf("%d n'est pas un nombre
25                             parfait",nb);
26                     break;
27                 default: return(0);
28             }
29         }while(true);
30
31         // Retourne 0 car le programme ne plante pas
32         return (0);
33     }

```