

ECUE «Introduction à la programmation »
 Session 2
 12 juin 2018
 sans document
 durée 1 heure 30

Exercice 1 (3 points)

Ecrire un programme `ex01.c` permettant à l'utilisateur d'entrer un nombre de victoires et un nombre de défaites et affichant le pourcentage de victoires. En particulier, la sortie du programme doit correspondre à l'exécution ci-dessous. On suppose que l'utilisateur entre des valeurs strictement positives. On affichera le pourcentage avec un chiffre après la virgule. Les entrées clavier sont indiquées en caractères gras.

```
nombre de victoires ? 15
nombre de defaites ? 25
% victoires = 37.5
```

Exercice 2 (5 points)

Donner la sortie du programme ci-dessous. Pour chaque ligne en **caractères gras**, tenir compte de la **couleur de votre copie (bleu, rouge, vert, jaune)** pour **valoriser a et b** avec les valeurs précisées dans le commentaire de la ligne.

```
#include <stdio.h>
int main() {
    int a = 3; // Couleur de la copie : Bleu:3, Rouge:4, Vert:5, Jaune:7
    int * p = &a;
    int b = *p; printf("1: a = %d, b = %d, *p = %d.\n", a, b, *p);
    a += 4; // Couleur de la copie : Bleu:4, Rouge:5, Vert:3, Jaune:4
    printf("2: a = %d, b = %d, *p = %d.\n", a, b, *p);
    b *= 7; // Couleur de la copie : Bleu:7, Rouge:3, Vert:3, Jaune:3
    printf("3: a = %d, b = %d, *p = %d.\n", a, b, *p);
    a -= 5; // Couleur de la copie : Bleu:5, Rouge:7, Vert:4, Jaune:5
    printf("4: a = %d, b = %d, *p = %d.\n", a, b, *p);
    b /= 3; // Couleur de la copie : Bleu:3, Rouge:4, Vert:5, Jaune:5
    printf("5: a = %d, b = %d, *p = %d.\n", a, b, *p);
    int * q = &b; printf("6: a = %d, b = %d, *p = %d, *q = %d.\n", a, b, *p, *q);
    *q *= (*p)++; printf("7: a = %d, b = %d, *p = %d, *q = %d.\n", a, b, *p, *q);
    *q -= ++(*p); printf("8: a = %d, b = %d, *p = %d, *q = %d.\n", a, b, *p, *q);
    p = q; printf("9: a = %d, b = %d, *p = %d, *q = %d.\n", a, b, *p, *q);
    q = &a; printf("10:a = %d, b = %d, *p = %d, *q = %d.\n", a, b, *p, *q);
    return 0;
}
```

Exercice 3 (4 points)

L'algorithme de Babylone calcule la racine carrée d'un nombre A avec une précision P . Il utilise une suite de nombres réels X_n tels que $X_{n+1} = (X_n + A/X_n) / 2$.

En C, programmer l'algorithme de Babylone en respectant les entrées sorties suivantes:

```
Calcul de la racine carree d'un nombre A avec une precision P.
Nombre A ? 2
Precision P ? 0.001
Valeur initiale ? 1.8
x1 = 1.455555, erreur = 0.344444
x2 = 1.414800, erreur = 0.040755
x3 = 1.414213, erreur = 0.000587
```

L'utilisateur entre le nombre A , la précision P et la valeur initiale X_0 de la suite. A chaque itération, le programme affiche la valeur de X_n et l'erreur $e = |X_n - X_{n-1}|$ avec 6 décimales. Le programme s'arrête lorsque l'erreur e est inférieure à P .

On pourra utiliser des variables a , p , x , $xsave$, e et n . Dans cet exercice, on n'utilisera pas de fonction, ni de tableau. (4 pts).

Exercice 4 (8 points)

1) Ecrire une procédure `void affiche` prenant en entrée un tableau t d'éléments de type `float` et un nombre n d'éléments et affichant les n premiers éléments du tableau sans chiffre après la virgule et séparés par un espace.

2) Ecrire une procédure `void init` prenant en entrée un tableau t d'éléments de type `float` et un nombre n d'éléments et initialisant le tableau avec des nombres aléatoires compris entre 0 et 9.

3) Ecrire une procédure `void tamovar` prenant en entrée un tableau t d'éléments de type `float` et un nombre n d'éléments et donnant en sortie la moyenne m et la variance v des n éléments du tableau. Rappel mathématique:

$$m = (\sum t[i]) / n \quad \text{et} \quad v = (\sum t[i]^2 - (\sum t[i])^2 / n) / n$$

4) Ecrire un programme `main` déclarant un tableau de taille 10 initialisé avec `init`, affichant le tableau avec `affiche`, calculant la moyenne et la variance avec `tamovar`. et affichant les résultats avec deux chiffres après la virgule. Le programme respectera la sortie suivante.

```
3 6 7 5 3 5 6 2 9 1
moyenne = 4.70, variance = 5.41
```