

ECUE MLI130B

«Introduction à la programmation»

Contrôle continu n°3 – 13 janvier 2014

sans document - durée 1 heure 30

Exercice 1 (4 points)

Ecrire un programme affichant un triangle de côté T . T est entré au clavier par l'utilisateur. Si $T < 2$, le programme affiche une erreur et retourne -1 . Sinon, il affiche le triangle avec des $*$ séparées par des $=$, puis retourne T . L'exécution du programme respectera les entrées-sorties ci-dessous.

T ? **4**
 ====*==*
 ==*====*==
 ===*==*===
 ====*=====

T ? **3**
 ====*
 ==*==*==
 ===*==*

T ? **2**
 ==
 ==*==

T ? **1**
 erreur: T<2.

Exercice 2 (5 points)

1) Ecrire une fonction `int diane(int a, int b)` demandant à l'utilisateur un nombre entier dans l'intervalle $[a, b]$ et retournant ce nombre. La fonction demande répétitivement le nombre à l'utilisateur tant que le nombre tapé au clavier n'appartient pas à $[a, b]$.

2) Ecrire une procédure `void correze` prenant en entrée deux nombres entiers, m et n , et deux nombres en sortie: le quotient q et le reste r de la division entière de m par n .

3) Ecrire un programme `main` demandant un entier c dans l'intervalle $[0, 99]$ et un entier d dans l'intervalle $[2, 9]$, puis affichant le quotient et le reste de la division entière de c par d . On utilisera la fonction `diane` et la procédure `correze` des questions 1 et 2. Le programme respectera les entrées sorties des exécutions suivantes.

Demande de c:
 x ? (0<=x<=99) **123**
 x ? (0<=x<=99) **-1**
 x ? (0<=x<=99) **45**
 c = 45
 Demande de d:
 x ? (2<=x<=9) **1**
 x ? (2<=x<=9) **6**
 d = 6
 45/6 = 7 reste 3

Demande de c:
 x ? (0<=x<=99) **60**
 c = 60
 Demande de d:
 x ? (2<=x<=9) **7**
 d = 7
 60/7 = 8 reste 4

Exercice 3 (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
```

Exercice 4 (3 points)

Pour chaque question, donner la sortie du programme.

1)

```
void cdt1(int t) { if (t==1) ; printf("A ");           printf("B "); }
void cdt2(int t) { if (t==1) printf("A ");           printf("B "); }
void cdt3(int t) { if (t==1) { printf("A "); return; } printf("B "); }
int main() {
    printf("0: "); cdt1(0); cdt2(0); cdt3(0);
    printf("\n1: "); cdt1(1); cdt2(1); cdt3(1); printf("\n"); return 0;
}
```

2)

```
void whl1(int t) { int i=t; while (i++<=3); printf("V "); printf("W "); }
void whl2(int t) { int i=t; while (i++<=3) printf("V "); printf("W "); }
void whl3(int t) { int i=t; while (i++<=3) { printf("V "); printf("W "); } }
int main() {
    printf("1: "); whl1(1); printf("\n2: "); whl2(1);
    printf("\n3: "); whl3(1); printf("\n"); return 0;
}
```

3)

```
void for1(int t) { int i=0; for (i=t; i<=3; i++) printf("F "); printf("G "); }
void for2(int t) { int i=0; for ( ; i<=3; i++) printf("F "); printf("G "); }
void for3(int t) { int i=0; for (i=t; ; i++) printf("F "); printf("G "); }
int main() {
    printf("1: "); for1(1); printf("\n2: "); for2(1);
    printf("\n3: "); for3(1); printf("\n"); return 0;
}
```