

ECUE MLI130B

«Introduction à la programmation»
 Contrôle continu n°3 – 13 janvier 2014

sans document - durée 1 heure 30

CORRIGE

CORRIGE

CORRIGE

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.

```
#include <stdio.h>
int main() {
    int T, i, j;
    printf("T ? "); scanf("%d", &T);
    if (T<2) {
        printf("erreur: T < 2.\n");
        return -1;
    }
    // 1 point:
    for (j=1; j<T; j++) printf("*=");
    printf("*\n");
    // 2 points:
    for (i=2; i<T; i++) {
        for (j=1; j<i; j++) printf("=");
        printf("*");
        for (j=i+1; j<=2*T-i-1; j++) printf("=");
        printf("*");
        for (j=1; j<i; j++) printf("=");
        printf("\n");
    }
    // 1 point:
    for (j=1; j<T; j++) printf("=");
    printf("*");
    for (j=1; j<T; j++) printf("=");
    printf("\n");
    return T;
}
```

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 $[a, b]$.

```
// 1 point
int diane(int a, int b) {
    int x;
    do {
        printf (" x ? (entre %d et %d) ", a, b);
        scanf("%d", &x);
    } while (x>b||x<a);
    return x;
}
```

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 .

```
// 2 points
void correze(int m, int n, int * q, int * r) {
    *q = m/n;
    *r = m%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
```

```
int main() {
    int c, d, q, r;
    // 0.5 point
    printf("Demande de c:\n"); c = diane(0, 99);
    printf("c = %d\n", c);
    printf("Demande de d:\n"); d = diane(2, 9);
    printf("d = %d\n", d);
    // 1 point
    correze(c, d, &q, &r);
    // 0.5 point
    printf("%d/%d = %d reste %d\n", c, d, q, r);
    return 0;
}
```

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.

```
// 1 point
void affiche(float * t, int l) {
    int i;
    for (i=0; i<l; i++)
        printf("%.0f ", t[i]);
    printf("\n");
}
```

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.

```
#define MODULO 10
// 1 point
void init(float * t, int l) {
    int i;
    for (i=0; i<l; i++) t[i]=rand()%MODULO;
}
```

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:

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

```
// 3 points
void tamovar(float * t, int l, float * mo, float * var) {
    int i; float s=0, s2=0;
    for (i=0; i<l; i++) {
        s += t[i];
        s2 += t[i]*t[i];
    }
    *mo = s/l;
    *var = (s2 - s*s/l)/l;
}
```

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
```

```
// 0.5 point
#define TAILLE 10
int main() {
    float tab[TAILLE];
    float m, v;
```

```
srand(1);

// 0.5 point
init(tab, TAILLE);
affiche(tab, TAILLE);

// 1.5 point
tamovar(tab, TAILLE, &m, &v);

// 0.5 point
printf("moyenne = %.2f, variance = %.2f\n", m, v);
return 0;
}
```

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;
}
```

```
// 1 point
0: A B B B
1: A B A B A
```

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;
}
```

```
// 1 point
1: V W
2: V V V W
3: V W V W V W
```

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;
}
```

```
// 1 point
1: F F F G
2: F F F F G
3: F F F F F F F F F F F F F F F F F F F F F F F F ... (boucle infinie)
```