

ECUE «Introduction à la programmation» - Session 2

12 juin 2013 - Bruno Bouzy

sans document - durée 1 heure 30

Exercice 1 (2 points)

Ecrire un programme `exo1.c` permettant à l'utilisateur d'entrer un nombre de tentatives et un nombre de succès et affichant le pourcentage de succès. La sortie du programme doit respecter la sortie ci-dessous. On suppose que l'utilisateur entre des valeurs strictement positives. On affichera le pourcentage avec un chiffre après la virgule. Les caractères gras sont en entrée du programme.

```

nombre de tentatives ? 200
nombre de tentatives = 200
nombre de succes ? 125
nombre de succes = 125
pourcentage de succes = 62.5

```

Exercice 2 (4 points)

Donner la sortie du programme suivant.

```

int main() {
    int a = 3; int * p = &a; int b = *p;
        printf("1: a = %d, b = %d, *p = %d.\n", a, b, *p);
    a *= 2;      printf("2: a = %d, b = %d, *p = %d.\n", a, b, *p);
    b += 4;      printf("3: a = %d, b = %d, *p = %d.\n", a, b, *p);
    int * q = &b; printf("4: a = %d, b = %d, *p = %d, *q = %d.\n", a, b, *p, *q);
    *q *= (*p)++; printf("5: a = %d, b = %d, *p = %d, *q = %d.\n", a, b, *p, *q);
    *q += ++(*p); printf("6: a = %d, b = %d, *p = %d, *q = %d.\n", a, b, *p, *q);
    p = q;      printf("7: a = %d, b = %d, *p = %d, *q = %d.\n", a, b, *p, *q);
    q = &a;      printf("8: a = %d, b = %d, *p = %d, *q = %d.\n", a, b, *p, *q);
    return(0);
}

```

Exercice 3 (6 points)

On considère les suites de nombres réels A_n et B_n définies de la manière suivante:

$$A_0=1 \quad B_0=9 \quad (3) \quad A_{n+1}=(A_n+B_n)/2 \quad B_{n+1}=(A_n B_n)^{1/2} \quad (4)$$

1) a) Quelles sont les valeurs de A_1 et B_1 ? **(0.25 pt)**

1) b) Donner la sortie de: **(0.5 pt)**

```
float a=1,b=9; a=(a+b)/2; b=sqrt(a*b); printf("a1=%.2f, b1=%.2f\n", a, b);
```

1) c) Cette sortie est-elle compatible avec la définition (4) ? **(0.25 pt)**

1) d) Modifier le traitement de la question 1) b) pour qu'il corresponde à la définition (4). **(1 pt)**

2) Ecrire une fonction `void deABaAB(float * a, float * b)` prenant a, b en entrée et les valorisant en sortie selon la définition (4). **(1.5 pt)**

3) Ecrire une fonction `void suite(int n)` affichant les n premiers termes des suites A_n et B_n avec 6 chiffres après la virgule. On utilisera `deABaAB`. **(2 pts)**

4) Ecrire un programme `main` appelant `suite` avec $n=3$. **(0.5 pt)**

Exercice 4 (8 points)

Soit `triRapide.c` le programme suivant.

```

#define TAILLE 8

void affiche(int * t, int premier, int dernier) {
    int i; printf("[ ");
    for (i=0; i<premier; i++) printf ("_ ");
    for (i=premier; i<=dernier; i++) printf ("%d ", t[i]);
    for (i=dernier+1; i<TAILLE; i++) printf ("_ ");
    printf("]\n");
}

int partition(int * t, int premier, int dernier) {
    int i, tmp, j = premier;
    for (i=premier; i<dernier; i++) {
        if (t[i] <= t[dernier]) {
            tmp = t[i]; t[i] = t[j]; t[j] = tmp; j++;
        }
        printf("P: i = %d j = %d ", i, j); affiche(t, premier, dernier);
    }
    tmp = t[dernier]; t[dernier] = t[j]; t[j] = tmp;
    printf("P: pivot = %d ", j); affiche(t, premier, dernier);
    return j;
}

void triRapide(int * t, int premier, int dernier, int niveau) {
    if (premier<dernier) {
        printf("TR debut: pre = %d der = %d niv = %d\n", premier, dernier, niveau);
        int pivot = partition(t, premier, dernier);
        triRapide(t, premier, pivot-1, niveau+1);
        triRapide(t, pivot+1, dernier, niveau+1);
        printf("TR fin: pre = %d der = %d niv = %d ", premier, dernier, niveau);
        affiche(t, premier, dernier);
    }
}

int main() {
    int tab[] = { 6, 25, 18, 0, 15, 10, 12, 16 };
    triRapide(tab, 0, TAILLE-1, 0);
    return 0;
}

```

1) Avec quelles valeurs de `premier`, `dernier` et `niveau`, `triRapide` est-elle appelée par `main` ? **(0.25 pt)**

2) a) Donner la sortie du programme jusqu'à la fin de la première exécution de `partition`. **(2 pts)**

2) b) En quelle position du tableau la dernière valeur du tableau a-t-elle été déplacée ? **(0.5 pt)**

2) c) Cette position s'appelle le pivot. Quelle est la valeur du tableau au pivot ? **(0.25 pt)**

2) d) Que vérifient les valeurs dont la position est inférieure à la valeur du pivot ? **(0.25 pt)**

2) e) Que vérifient les valeurs dont la position est supérieure à la valeur du pivot ? **(0.25 pt)**

3) a) Avec quelles valeurs de `premier` et `dernier`, `triRapide` de niveau 1 est-elle appelée la première fois par `triRapide` de niveau 0 ? la seconde fois ? **(0.5 pt)**

3) b) Donner la suite de la sortie du programme jusqu'à la fin de l'exécution du `main`. **(2.5 pts)**

3) c) Dessiner l'arbre d'appel des fonctions `triRapide` et `partition`. Chaque noeud de l'arbre correspondra à un appel de `triRapide` pour lequel `premier<dernier` est vrai ou à un appel de `partition`. Dans chaque noeud, on précisera les valeurs de `premier`, `dernier`. **(1.5 pt)**