Université Paul Sabatier
Partiel Programmation impérative (L2)
Mars 2005-03-27
2H- Documents autorisés

I- Sous-programmes : passage de paramètres

Q1. Considérons le programme suivant : **(2pts)**

```
int T[5] ; int i ;
void modif (int *x )
{ i = i+1 ; *x = *x +1 ; }
main()
        {int j;
        for (j = 0 ; j<5 ; j = j+1) T[j] = 0 ;
        i = 2 ;
        modif ( &(T[i]) ) ;
        printf(''i = %d\n'',  i);
        for (j = 0 ; j<5 ; j = j+1) printf(''%d'',  T[j]);
        }
```

Quelles sont les valeurs affichées?

i = 3

0 0 1 0 0

Q2. (2pts)

```
typedef int tab[2] ;
tab A ;
void farfelu ( tab t)
 { A[0] = t[0] + t[1]; A[1] = t[0] + t[1];}
```

```
main()
        { A[0] = 1 ; A[1] = 1 ; farfelu(A) ; printf(''%d   %d'', A[0], A[1]); }
```

Quelles sont les valeurs affichées par le programme?

2 3

II- Sous programme : construction

Q1. (3pts)

Ecrire un sous-programme en C qui reçoit en entrée un tableau de N entiers (N>0).
Ce sous-programme calcule :
La moyenne entière des valeurs du tableau
Affecte à chaque case du tableau la valeur 0 si elle contenait une valeur inférieure à la moyenne calculée sinon on affecte 1.

```
#deine N10
typedef int tab[N] ;
void binariser ( tab t)
        { int i , som, m ;
        som = 0;
        for (i = 0 ; i< N ; i=i+1) som = som + t[i];
        m = som/N;
        for (i = 0 ; i< N ; i=i+1) if  (t[i] < m) t[i] = 0; else t[i] = 1;
        }
```

Q2. Ecrire la fonction main qui utilise le sous-programme de la question précédente. **(2pts)**

```
main()
        { tab t;
                for (i = 0 ; i< N ; i=i+1) scanf("%d", &t[i]);
                binariser (t);
                for (i = 0 ; i< N ; i=i+1) printf("%d", t[i]);
}
```

II- Spécification.

1. Considérons le programme suivant :

```
typedef int vect[10] ;
/* la fonction init_v initialise un vecteur de type vect par des entiers strictement positifs */
void init_v ( vect v) { int i; for (i=0; i<10; i=i+1) {scanf(''%d'', &v[i]);}
int p (int a, int b)
 { /* a,b >0*/
          int x = a;
          while (x%b != 0) x = x + a;
return x;
/* x %a = 0 ∧ x%b=0 ∧ (∀Y : 0 < Y ∧  Y%a=0 ∧ Y%b=0 → Y>x */ }

void print_v ( vect v) { int i; for (i=0; i<10; i=i+1) {printf(''%d'', v[i]);};
main()
          {         vect v1;
                    int i, x;
                    init (v1);

/* A est un tableau qui représente les valeurs initiales de v1 */
/* v1 = A et chaque case de v1 contient un entier strictement positif*/
                    x = v1[0];
                    for ( i=0; i<9; i = i+1) v1 [i] = p(v1[i], v1[i+1]);
                    v1[9] = p(v1[9], x);
/* prédicat de sortie:…….*/
                    print_v(v1) ;
}
```

Q1. Soit : (1.5pts)

x = p(1, 3) ; **3**
y = p(2,3) ; **6**
z = p(8,12) ; **24**

Donnez les valeurs de x, y et z.

Q2. Que représente la valeur de retour de la fonction p par rapport à ses deux valeurs d'entrée ? **(0.5 pt)**

Le plus petit commun multiple

Q3. Donner une représentation informelle du prédicat de sortie présenté en commentaire dans le main() **(2pts)**

Chaque case v1[i] contient le ppcm de (A[i], A[(i+1)%N) avec 0<= i <= N-1

Q4. Ecrire les formules qui correspondent aux prédicats d'entrée et de prédicat de sortie présentés en commentaires dans le main(). **(4pts)**

/* ($\forall$I : 0 ≤ I ≤N-1 $\rightarrow$ v[I] >0) $\wedge$ N>0 */

```
x = v1[0];
for ( i=0; i<9; i = i+1) v1 [i] = p(v1[i], v1[i+1]);
v1[9] = p(v1[9], x);
```

/* ($\forall$I : 0 ≤ I ≤N-1 $\rightarrow$ v[I] = p(A[I], A[(I+1) % N]) */

2. Ecrire la spécification d'un programme qui reçoit en entrée un tableau T de N entiers qui sont différents entre eux avec (N>0). Ce programme fournit en sortie les indices deb et fin qui délimitent un sous-tableau T[deb..fin] de taille la plus grande de T qui soit trié dans l'ordre croissant.

Nous pouvons utiliser le prédicat :

TRIE(T(A..B), ≤) pour dénoter la formule : ($\forall$I : A ≤ I < B $\rightarrow$ T[I] ≤ T[I+1])

Q1. Calculer dans **longueur** la taille du tableau T(deb..fin) **(0.5 pt)**

longueur = fin-deb + 1

Q2. Formaliser : tous les sous tableaux de T de taille = long + 1 ne sont pas triés **(1 pt)**

($\forall$I : 0 ≤ I ≤ N-long -1 $\wedge$ I # deb $\rightarrow$ ¬TRIE(T(I..I+long))

Q3. Ecrire la spécification complète **(1.5pts)**

/* ($\forall$I : 0 ≤ I<J ≤N-1 $\rightarrow$ T[I] # T[J]) $\wedge$ N>0 */

Sous_tableau_decr_max(T, &deb, &fin) ;

/* (0 ≤ deb ≤ fin ≤ N-1) ∧ TRIE(T[deb..fin], ≤) ∧ (∀I : 0 ≤ I ≤ N-long -1 → ⌉TRIE(T [I..I+long]) */