

Examen de I1
Session de janvier 2007

Date : le lundi 8 janvier
Durée : 1heure 30

EPREUVE SANS DOCUMENTS
SANS CALCULATRICES

1. Qu'est-ce qu'une "variable" ? Soient x et y deux variables de type "entier", que réalise l'instruction suivante :

x = y;

(que se passe-t-il exactement en mémoire d'ordinateur ?).

CORRECTION : copie du contenu de la zone mémoire identifiée par y dans la zone mémoire identifiée par x

2. Soient x et y deux variables de type "int", Les instructions suivantes sont-elles légales (possibles) et pourquoi? *CORRECTION :*

- $x = -x$; *oui affecte à x le négatif de sa valeur*
- $2x = y$; *non, un identificateur ne commence pas par 1 chiffre*
- $y = 2*x$; *oui, affecte à y le résultat de 2 * le contenu de x*
- $2 = 2-x$; *non, un identificateur ne commence pas par 1 chiffre*
- $y-2 = x-1$; *non, à gauche de = il doit y avoir 1 variable seule*
- $x = 2$; *oui affecte à y la valeur de 2*

3. Qu'appelle-t-on type composé par opposition à la notion de type simple? Donnez 2 exemples de variables de type composé.

CORRECTION : Un type simple est tel que toute variable de ce type ne possède, à 1 instant t, qu'une et une seule valeur. Une variable de type composé peut posséder plusieurs valeurs (de type simple !) à 1 instant t. Exemple : un tableau d'entiers, une struct composée de champs, une chaîne de caractères.

4. Sachant que les variables a, b et c ont respectivement pour valeur 2, 8 et 6, déterminer si les tests suivants sont vrai ou faux :

- | | |
|---|----------------------------------|
| ▪ $a == b$ | <i>faux</i> |
| ▪ $!(b != 0)$ | <i>faux</i> |
| ▪ $(b - a) == c$ | <i>vrai</i> |
| ▪ $((2*a + c) >= b) \ \&\& \ (a <= (c - b))$ | <i>(v ET f) = faux</i> |
| ▪ $((c - a) < b) \ \ ((a + c) > b)$ | <i>(v OU f) = vrai</i> |
| ▪ $!(!(b < c) \ \&\& \ ((a < c) \ \ (a > b)))$ | <i>NON(v ET (v OU f)) = faux</i> |

5. Traduire en langage C les déclarations des variables suivantes: *CORRECTION :*

- | | |
|---|------------------------------|
| ▪ x : entier long | <i>long int x;</i> |
| ▪ mot : "chaîne" de 30 caractères, | <i>char mot[30];</i> |
| ▪ liste : tableau de 15 entiers courts, | <i>short int liste[15];</i> |
| ▪ matrix :tableau de 10 tableaux de 20 réels, | <i>float matrix[10][20];</i> |

6. La fonction C suivante est censée calculer et retourner la somme s des carrés de tous les diviseurs d'un entier n (sauf lui-même). Pourtant, l'exécution montre des résultats erronés. Trouvez et corrigez les 4 erreurs de logique qui se sont glissées dans cet algorithme:

```
int somdiv2(int n)
{
    int i, s;
```

```

s = 0; CORRECTION : s=1;
for (i=2; i<=n; i=i+1) CORRECTION: for (i=2; i<n; i=i+1)
    if ((n%i) != 0) CORRECTION: if((n%i)==0) s=s+i*i;
return s;
}

```

7. Que réalise la fonction C suivante :

```

void faitquoi(int k);
{ int i, j;
  for (j = 2; j <= k ; j = j+1)
    { i=2;
      while ((i<j) &&(j%i!=0))
        i = i+1;
      if (i == j)printf ("%d \n", j);
    }
}

```

CORRECTION : Affiche tous les nombres premiers entre 2 et k (compris)

8. Soit les instructions suivantes :

```

#define N 50
typedef int VECTEUR [N];

```

8.a. Ecrire la fonction C permettant de tester si son paramètre de type VECTEUR est trié ou pas. La fonction retournera 1 si le vecteur est trié et 0 sinon.

```

CORRECTION :
int trie(VECTEUR v)
{ int i=0;
  while ((i<N-1) && (v[i]< v[i+1]))
    i = i+1;
  if (i< N-1) return 0;
  else return 1;
}

```

8.b. Soit un VECTEUR v trié par ordre croissant (et ne possédant pas de doublons), écrire la fonction C de recherche dichotomique d'un entier m (fourni en paramètre) dans ce VECTEUR v (en paramètre également). La fonction retournera:

- Soit l'indice de position de m dans le vecteur, si m ∈ au vecteur
- Soit -1 si m ∉ au vecteur

```

CORRECTION :
int rechDicho(VECTEUR v, int m)
{ int deb=0, fin=N-1, milieu;
  milieu = (deb+fin)/2;
  while ((deb<=fin) &&(v[milieu] !=m))
  {
    if (m < v[milieu]) fin = milieu -1;
    else deb = milieu +1;
    milieu = (deb+fin)/2;
  }
  if (v[milieu] == m) return milieu;
  else return -1;
}

```

8.c. Expliquer le principe général du tri par sélection appliqué sur un VECTEUR v.

Pour tout j de 0 à N-1

Faire pour tout i de j+1 à N-1

Faire rechercher le MIN et son indice k

Fait
Permuter $v[k]$ et $v[j]$
fait