

Interrogation du 05/10/2009

Corrigé

Exercice I

1.

```
int est_bisextile (int annee)
/* Retourne 1 si l'annee annee est bisextile et 0 sinon */
{
    return (((annee%4==0)&&(annee%100!=0))
            /* Annee divisible par 4 et non divisible par 100 */
            || (annee%400==0)); /* ou annee divisible par 400 */
}
```

2.

```
int jours_dans_mois (int mois, int annee)
/* Retourne le nombre de jours dans le mois mois/annee */
{
    /* Mois de 30 jours */
    if ( ((mois%2==0) && (mois<8) && (mois!=2))
        || ((mois%2==1) && (mois>=8)) )
        return 30;

    /* Mois de 31 jours */
    else if ( ((mois%2==1) && (mois<=7))
             || ((mois%2==0) && (mois>=8)) )
        return 31;

    /* Le cas particulier du mois de fevrier */
    else if (mois==2) {
        if (est_bisextile(annee)) return 29;
        else return 28;
    }
}
```

3.

```
typedef struct date { int jour, mois, annee; } DATE;
```

4. Plusieurs méthodes sont envisageables.

Une première méthode consiste à convertir une date en un nombre de jours avant ou après une date de référence. La différence de dates se fait alors par la différence des deux entiers.

Une seconde méthode, présentée au verso, consiste à simplement compter le nombre de jours, comme une personne le ferait sur un calendrier. Cette méthode a l'avantage de la simplicité de la programmation mais l'inconvénient d'être moins rapide, en particulier si les deux dates sont éloignées.

```

int date_difference (DATE d1, DATE d2)
/* Calcule la difference entre deux dates
   ENTREE : d1 et d2 les deux dates
   SORTIE : le nombre de jours entre d1 et d2
*/
{
    if ((d1.annee>d2.annee) ||
        ((d1.annee==d2.annee) && (d1.mois>d2.mois)) ||
        ((d1.annee==d2.annee) && (d1.mois==d2.mois) && (d1.jour>d2.jour)))

        /* Si d2 est avant d1 */
        return -date_difference(d2,d1);

    else {
        /* Si d1 est avant d2 */
        int resultat = 0;
        DATE d = d1;

        while ((d.jour!=d2.jour)|| (d.mois!=d2.mois)|| (d.annee!=d2.annee)) {
            /* On incremente resultat... */
            resultat++;

            /* ... et on incremente d */
            d.jour++;
            if (d.jour>jours_dans_mois(d.mois,d.annee)) { d.jour=1; d.mois++; }
            if (d.mois>12) { d.mois=1; d.annee++; }
        }

        return resultat;
    }
}

```

Exercice II

1.

```

int is_coprime(int n1, int n2)
/* Predicat de primalite relative de deux entiers n1 et n2
   ENTREE : n1 et n2
   SORTIE : 1 si n1 et n2 ont un facteur autre que 1 en commun, 0 sinon
*/
{
    int i; /* compteur de boucle pour les diviseurs */
    int PremiersEntreEux = 1; /* passera a 0 si un contre exemple est trouve */
    int min = (n1<n2)?n1:n2; /* min de n1 et n2 */

    for (i=2;i<=min;i++)
        /* Si n1 et n2 ont un facteur commun, ils ne sont pas premiers entre eux */
        if ((n1%i==0)&&(n2%i==0)) PremiersEntreEux = 0;

    return PremiersEntreEux;
}

```

2.

```
int phi(int n)
/* Fonction Indicatrice d'Euler
   ENTREE : n
   SORTIE : Phi(n)
*/
{
    int i;
    int resultat=0;

    for (i=1;i<=n;i++)
        if (is_coprime(i,n))
            resultat++;

    return resultat;
}
```

3.

```
#include <stdio.h>

/* Inclure les fonctions des questions 1 et 2 ici */

int verifierconjecture (int n)
/* Verification de la conjecture de Carmichael pour n
   ENTREE : n
   SORTIE : 1 s'il existe un m tel que phi(n)=phi(m), 0 sinon
   REMARQUE : on arrete de tester pour m>=nmax (1000000)
*/
{
    int m=1;
    int mmax = 1000000;

    while ( ((phi(n)!=phi(m)) || (n==m)) && (m<mmax) )
        m++;

    return(phi(n)==phi(m));
}

int main() {
    int conjectureok = 1;
    int n;

    for (n=1;n<=10000;n++)
        /* Verification de la conjecture pour les 10000 premiers entiers non nuls */
        conjectureok = conjectureok && verifierconjecture(n);

    if (conjectureok) printf("Conjecture verifiee entre 1 et 10000\n");
    else printf("Conjecture fausse\n");
}
```

4. Soit $n \geq 2$ un entier. L'entier n est premier équivaut à tous les entiers entre 1 et $n - 1$ sont premiers avec n ce qui équivaut à $\varphi(n) = n - 1$. Pour répondre à la question, on remplace le main de la question précédente par le main suivant

```
int main() {
    int n;

    for (n=2;n<=100;n++) {

        /* Calcul de phi(n) */
        int phin = phi(n);
        printf("n=%d, phi(n)=%d, ",n,phin);

        /* Primalite */
        if (phin==n-1) printf("n est premier\n");
        else printf("n n'est pas premier\n");
    }

    return 0;
}
```

Exercice III

Soit N les 14 chiffres constituant les trois premières composantes, on a

$$N = N_{3b} + 10^5 N_{3a} + 10^{10} N_2 + 10^{13} N_1$$

alors $N \bmod 7$ et égal à

$$(N_{3b} \bmod 7 + (10^5 \bmod 7)(N_{3a} \bmod 7) + (10^{10} \bmod 7)(N_2 \bmod 7) + (10^{13} \bmod 7)(N_1 \bmod 7)) \bmod 7$$

alors

$$N \bmod 7 = (N_{3b} \bmod 7 + 5N_{3a} \bmod 7 + 4N_2 \bmod 7 + 3N_1 \bmod 7) \bmod 7$$

```
#include <stdio.h>
```

```
int main() {
    int N1, N2, N3a, N3b, N4;
    int ClefCalculee;

    printf("Entrer le numero du voyage :\n"); scanf("%d",&N1);
    printf("Entrer le numero de la compagnie :\n"); scanf("%d",&N2);
    printf("Entrer les 5 premiers chiffres du numero du document :\n"); scanf("%d",&N3a);
    printf("Entrer les 5 derniers chiffres du numero du document :\n"); scanf("%d",&N3b);
    printf("Entrer la clef :\n"); scanf("%d",&N4);

    ClefCalculee = ( N3b%7 + 5*N3a%7 + 4*N2%7 + 3*N1%7 ) % 7 ;

    if (N4==ClefCalculee) printf("Numero valide\n");
    else printf("Numero invalide, il devrait se terminer par %d\n",ClefCalculee);

    return 0;
}
```