Auteur : Mohamed Messabihi
Matière : Algorithmique et structures de données II
Date 17 avril 2022
Durée 1h30

Université Abou Bakr Belkaïd - Tlemcen
Faculté des Sciences
1ère Année MI
Semestre 2

Contrôle Continu

Aucun document n'est autorisé
Les solutions doivent être rédigées en **C**
Les appareils portables doivent être éteints et posés sur le bureau du surveillant

1 Affichage — 10 pts. 40'

Qu'affichent les deux programmes suivants :

```c
#include<stdio.h>
void Toto(int *a, int *b, int *c)
{   *a=*b;
    *b=*c;
    *c=*a;
}
void Loulou(int *a, int *b, int *c)
{   int *d=a;
    a=b;
    b=c;
    c=d;
}
void main()
{   int i=0, j=1, k=2;
    printf("%d %d %d\n",i, j, k);
    Toto(&i, &j, &k);
    printf("%d %d %d\n",i, j, k);
    i=0, j=1, k=2;
    Loulou(&i, &j, &k);
    printf("%d %d %d\n",i, j, k);
}
```

```c
#include<string.h>
void main()
{
 int T[5]={1, 2, 3, 4, 5};
 int A[5]={1};
 int i, *p = T, *q=A;
 printf("T[0] = %d \n", T[0]);
 printf("A[1] = %d \n", A[1]);
 printf("*p=%d, *q=%d \n", *p, *q);
 q++;
 printf("*q = %d \n", *q);
 for(i=1;i<5;i++)
   {
     p++;
     (*q)= *(p);
     q++;
     printf("A[i]=%d\n",A[i]);
   }
 printf("*p++p-- = %d \n", *p++p--);
 printf("*p++--p = %d \n", *p++--p);
}
```

2 Nombres aimables — (10 pts, 50')

En arithmétique, deux nombres (entiers strictement positifs) sont dits aimables s'ils sont distincts et si chacun des deux nombres est égal à la somme des diviseurs stricts de l'autre. On rappelle que les diviseurs stricts d'un entier naturel n sont tous les entiers naturels diviseurs de n sauf n lui même (par exemple, Les diviseurs stricts de 30 sont : 1, 2, 3, 5, 6, 10, et 15).

Si l'on note $s(n)$ la somme des diviseurs stricts de n, deux nombres distincts m et n sont donc aimables si et seulement si :

$$s(m) = n \quad \text{et} \quad s(n) = m$$

Par exemple, les nombres entiers 220 et 284 sont aimables car :

— s(220) = 1+2+4+5+10+11+20+22+44+55+110 = **284**
— s(284) = 1+2+4+71+142 = **220**

1. Écrire une fonction somme_diviseurs qui retourne la somme des diviseurs stricts d'un nombre passé en paramètre (**4 pts**).
2. Écrire une fonction mon_aimable qui retourne le nombre aimable (s'il existe) d'un nombre passé en paramètre (**3 pts**).
3. Écrire un programme principal qui affiche tous les couples de nombres aimables inférieurs à une certaine limite donnée par l'utilisateur (**3 pts**).

« Bon courage »

Auteur : Mohamed Messabihi
Matière : Algorithmique et structures de données II
Date 17 avril 2022
Durée 1h30

Université Abou Bakr Belkaïd - Tlemcen
Faculté des Sciences
1ère Année MI
Semestre 2

Correction du Contrôle Continu

Aucun document n'est autorisé
Les solutions doivent être rédigées en **C**
Les appareils portables doivent être éteints et posés sur le bureau du surveillant

1 Affichage — 10 pts. 40'

Qu'affichent les deux programmes suivants :

```c
#include<stdio.h>
void Toto(int *a, int *b, int *c)
{   *a=*b;
    *b=*c;
    *c=*a;
}
void Loulou(int *a, int *b, int *c)
{   int *d=a;
    a=b;
    b=c;
    c=d;
}
void main()
{   int i=0, j=1, k=2;
    printf("%d %d %d\n",i, j, k);
    Toto(&i, &j, &k);
    printf("%d %d %d\n",i, j, k);
    i=0, j=1, k=2;
    Loulou(&i, &j, &k);
    printf("%d %d %d\n",i, j, k);
}
```

```c
#include<string.h>
void main()
{
 int T[5]={1, 2, 3, 4, 5};
 int A[5]={1};
 int i, *p = T, *q=A;
 printf("T[0] = %d \n", T[0]);
 printf("A[1] = %d \n", A[1]);
 printf("*p=%d, *q=%d \n", *p, *q);
 q++;
 printf("*q = %d \n", *q);
 for(i=1;i<5;i++)
   {
     p++;
     (*q)= *(p);
     q++;
     printf("A[i]=%d\n",A[i]);
   }
 printf("*p++p-- = %d \n", *p++p--);
 printf("*p++--p = %d \n", *p++--p);
}
```

Solution

Affichage

```
0 1 2
1 2 1
0 1 2
```

Affichage

```
T[0] = 1
A[1] = 0
*p=1, *q=1
*q = 0
A[i]=2
A[i]=3
A[i]=4
A[i]=5
*p+*p- - = 10
*p+*- -p = 7
```

2 Nombres aimables — (10 pts, 50')

En arithmétique, deux nombres (entiers strictement positifs) sont dits aimables s'ils sont distincts et si chacun des deux nombres est égal à la somme des diviseurs stricts de l'autre. On rappelle que les diviseurs stricts d'un entier naturel n sont tous les entiers naturels diviseurs de n sauf n lui même (par exemple, Les diviseurs stricts de 30 sont : 1, 2, 3, 5, 6, 10, et 15).

Si l'on note $s(n)$ la somme des diviseurs stricts de n, deux nombres distincts m et n sont donc aimables si et seulement si :

$$s(m) = n \quad \text{et} \quad s(n) = m$$

Par exemple, les nombres entiers 220 et 284 sont aimables car :

— s(220) = 1+2+4+5+10+11+20+22+44+55+110 = **284**

— s(284) = 1+2+4+71+142 = **220**

1. Écrire une fonction somme_diviseurs qui retourne la somme des diviseurs stricts d'un nombre passé en paramètre (**4 pts**).

Solution

```c
int somme_diviseurs ( int nombre )
{
        int i, somme = 0;
        for (i=1; i<=nombre/2; i++)
        {
                if (nombre % i == 0)
            somme = somme + i;
        }
        return somme;
}
```

2. Écrire une fonction mon_aimable qui retourne le nombre aimable (s'il existe) d'un nombre passé en paramètre (**3 pts**).

Solution

```c
int mon_aimable( int nombre )
{
        int mon_aimable;
        mon_aimable = somme_diviseurs(nombre);
        if (somme_diviseurs(mon_aimable) == nombre)
                return mon_aimable;
        else
                return nombre;
}
```

3. Écrire un programme principal qui affiche tous les couples de nombres aimables inférieurs à une certaine limite donnée par l'utilisateur (**3 pts**).

Solution

```c
#include <stdio.h>
/* calcul de la somme des diviseurs du parametre nombre */
int somme_diviseurs( int nombre );
/* calcul de l'aimable du parametre nombre */
int mon_aimable( int nombre );

int main ( )
{
        int nombre, limite, aimable;
        printf("Limite : ");
        scanf("%d", &limite);
        for (nombre=1; nombre<=limite; nombre++)
        {
                aimable = mon_aimable(nombre);
                if (aimable != nombre)
                        printf("%d et %d sont aimables\n", nombre, aimable);
        }
        return 0;
}
```

« Bon courage »