

ĐỀ KIỂM TRA GIỮA KỲ # HK1 – 2021-2022

Môn: **Nhập môn Lập trình** - Lớp: **21CLC5** - GV: **Thái Hùng Văn**

Thời gian làm bài: **75 phút** - **Không** dùng tài liệu, ĐTDĐ, bàn phím Laptop /Desktop PC

Câu 1: Bài toán “100 trâu 100 cỏ, trâu đực ăn 5, trâu nái ăn 3, lụ khụ trâu già, 3 con ăn 1” có 1 hàm xử lý như sau:

```
1 bool TimSoTrau(int &SoTD, int &SoTN, int &SoTG, int TongSoTrau=100, int TongSoCo=100, int min=0)
2 {
3     for (SoTD = min; SoTD <= TongSoCo/5; SoTD++) { // SoTD == so trau dung
4         int SoTN_TG = TongSoTrau - SoTD;           // SoTN_TG == so trau nam + so trau gia
5         for (SoTG = 0; SoTG <= SoTN_TG; SoTG+=3) { // SoTG == so trau gia
6             SoTN = SoTN_TG - SoTG;                 // SoTN == so trau nam
7             if (SoTD * 5 + SoTN * 3 + SoTG / 3 == TongSoCo)
8                 return true;
9         }
10    }
11    return false;
12 }
```

Hãy viết hàm main gọi hàm trên sao cho có thể xuất ra **tất cả các nghiệm** của bài toán.

Câu 2: Vẽ lưu đồ (Flow Chart) xác định ước chung lớn nhất của hai số nguyên dương. (Vd, UCL2 của 18 và 24 là 3)

Câu 3: Gọi **thời_điểm** là một mốc giờ:phút:giây cụ thể trong ngày và **thời_gian** là số giây giữa 2 **thời_điểm**. (vd, **thời_điểm** bắt đầu làm bài thi là 14:00:01 và **thời_điểm** kết thúc là 15:00:02 thì **thời_gian** thi là 3601 giây). Hãy viết các hàm thực hiện các việc sau :

a/ Xác định **thời_gian** đã trôi qua kể từ **00:00:00** đến một **thời_điểm** nào đó.

(vd, **thời_gian** đã trôi qua từ 0 giờ đến **thời_điểm** 01:02:03 là 3723 giây)

b/ Xác định **thời_gian** đã trôi qua giữa 02 **thời_điểm**

c/ Xác định **thời_điểm** khi biết **thời_gian** đã trôi qua kể từ **00:00:00**.

d/ Xác định 2 **thời_điểm** gần nhau nhất trong 3 **thời_điểm** A, B, C

--HẾT--

Bài Sửa

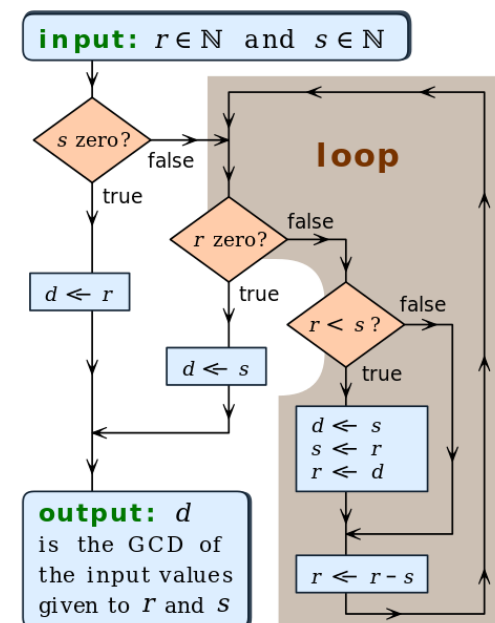
CÂU 1

```
main.cpp
1 #include <iostream>
2 using namespace std;
3 bool TimSoTrau(int &SoTD, int &SoTN, int &SoTG, int TongSoTrau=100, int TongSoCo=100, int min=0) {
4     for (SoTD = min; SoTD <= TongSoCo/5; SoTD++) { // SoTD == so trau dung
5         int SoTN_TG = TongSoTrau - SoTD; // SoTN_TG == so trau nam + so trau gia
6         for (SoTG = 0; SoTG <= SoTN_TG; SoTG+=3) { // SoTG == so trau gia
7             SoTN = SoTN_TG - SoTG; // SoTN == so trau nam
8             if (SoTD * 5 + SoTN * 3 + SoTG / 3 == TongSoCo)
9                 return true;
10        }
11    }
12    return false;
13 }
14 int main() {
15     int SoCon=100, SoCo=100;
16     do {
17         int td, tn, tg;
18         cout << "Neu tong so trau la "<<SoCon<<" va tong so co la "<< SoCo;
19         for (int i=0, min=0; TimSoTrau(td, tn, tg, SoCon,SoCo,min); i++,min=td+1)
20             cout << "\n * So trau 3 loai - truong hop "<<i+1<<": " << td << " " << tn << " " << tg;
21         cout << "\n\n Nhap tong so trau va tong so co moi (nhap 0 neu ko lam tiep): ";
22         cin >> SoCon >> SoCo;
23     } while (SoCon*SoCo);
24 }
```

input

```
Nhap tong so trau va tong so co moi (nhap 0 neu ko lam tiep): 1000 2020
Neu tong so trau la 1000 va tong so co la 2020
 * So trau 3 loai - truong hop 1 : 2 629 369
 * So trau 3 loai - truong hop 2 : 6 622 372
 * So trau 3 loai - truong hop 3 : 10 615 375
```

CÂU 2: (https://fr.m.wikipedia.org/wiki/Fichier:Academ_Flowchart_to_get_simply_the_GCD_of_two_positive_integers.svg)



CÂU 3

* **Prototype 4 hàm của 4 câu như sau :**

int ThoiGian (int hh, int mm, int ss) ; // hàm trả về **số giây** đã trôi qua từ thời_điểm **00:00:00** đến **hh:mm:ss**

int ThoiGian (int h1, int m1, int s1, int h2, int m2, int s2) ; // hàm trả về **số giây** từ thời_điểm **h1:m1:s1** đến **h2:m2:s2**

void ThoiDiem (int seconds, int &hh, int &mm, int &ss) ; // Xác định thời_điểm **hh:mm:ss** khi biết **số giây** kể từ 0 giờ

int GanNhat (int h1, int m1, int s1, int h2, int m2, int s2, int h3, int m3, int s3) ; // Xác định 2 thời_điểm gần nhau nhất trong 3 thời_điểm **T1, T2, T3**. Giá trị trả về của hàm là 12 khi kết quả là **T1 & T2**, là 13 khi kết quả là **T1 & T3**, là 23 khi kết quả là **T2 & T3**

* **Thân hàm :**

```
int ThoiGian ( int hh, int mm, int ss ) {  
    return hh*3600 + mm*60 + ss ;  
}
```

```
int ThoiGian ( int h1, int m1, int s1 , int h2, int m2, int s2) {  
    int ss1 = ThoiGian (h1, m1, s1) ; // số giây từ 0 giờ đến th/đ h1:m1:s1 (cũng có thể coi là th/đ theo giây)  
    int ss2 = ThoiGian (h2, m2, s2) ; // số giây từ 0 giờ đến th/đ h2:m2:s2 (cũng có thể coi là th/đ theo giây)  
    if (ss2 > ss1 )  
        return ss2 – ss1 ;  
    return ss1 - ss2 ;  
}
```

```
void ThoiDiem ( int seconds, int &hh, int &mm, int &ss) {  
    hh = seconds / 3600 ; // do 1 giờ = 3600 giây  
    int s = seconds % 3600 ; // số giây đã trôi qua tính từ đầu giờ  
    mm = s / 60 ;  
    ss = s % 60;  
}
```

```
int GanNhat ( int h1, int m1, int s1, int h2, int m2, int s2, int h3, int m3, int s3 ) {  
    int t12 = ThoiGian (h1, m1, s1, h2, m2, s2) ; // số giây từ giữa 2 th/đ T1 & T2  
    int t13 = ThoiGian (h1, m1, s1, h3, m3, s3) ; // số giây từ giữa 2 th/đ T1 & T3  
    int t23 = ThoiGian (h2, m2, s2, , h3, m3, s3) ; // số giây từ giữa 2 th/đ T2 & T3  
    if (t12< t13 && t12<t23) return 12 ;  
    if (t13< t12 && t13<t23) return 13 ;  
    return 23;  
}
```

* CT minh họa việc dùng các hàm trên:

```
int main ( ) {  
    cout << "\n So giay tu 0 gio den 7:30 la : " << ThoiGian (7, 30, 0) ;  
    cout << "\n So giay tu 7:30 den 9:10 la : " << ThoiGian (7,30,0, 9,10,0) ;  
    int h, m, s, t = 2021 ;  
    ThoiDiem (t, h, m, s) ;  
    cout << "\n Thoi diem " << t << " giay (ke tu 0 giờ) la : " << h << ":" << m << ":" << s ;  
    cout << "\n Voi 3 th/d T1=12:34:56, T2=01:02:03, T3=21:00:00 thi 2 th/d gan nhat la : "  
        << GanNhat (12,34,56, 1,2,3, 21,0,0) ;  
}
```

@ Nếu đã học kiểu Struct thì nên làm như sau :

* Định nghĩa kiểu ThoiDiem :

```
struct THOIDIEM {  
    int hh, mm, ss ;  
} ;
```

* Prototype 4 hàm :

int ThoiGian (THOIDIEM T) ; // hàm trả về số giây đã trôi qua từ thời_điểm 00:00:00 đến thời_điểm T

int ThoiGian (THOIDIEM T1, THOIDIEM T2) ; // hàm trả về số giây từ thời_điểm T1 đến T2

THOIDIEM ThoiDiem (int seconds) ; // Xác định thời_điểm khi biết số giây kể từ 0 giờ

int GanNhat (THOIDIEM T1, THOIDIEM T2, THOIDIEM T3) ; // Xác định 2 thời_điểm gần nhau nhất trong 3 thời_điểm T1, T2, T3. Giá trị trả về của hàm là 12 khi kết quả là T1 & T2, là 13 khi kq là T1 & T3, là 23 khi kq là T2 & T3

* Thân hàm :

```
int ThoiGian ( THOIDIEM T) {  
    return T.hh*3600 + T.mm*60 + T.ss ;  
}
```

```
int ThoiGian ( THOIDIEM T1, THOIDIEM T2) {  
    int ss1 = ThoiGian (T1) ; // số giây từ 0 giờ đến th/đ T1  
    int ss2 = ThoiGian (T2) ; // số giây từ 0 giờ đến th/đ T2
```

```
if (ss2 > ss1 ) return ss2 – ss1 ;
```

```
return ss1 - ss2 ;
```

```
}
```

```
THOIDIEM ThoiDiem ( int seconds) {
```

```
    THOIDIEM T ;
```

```
    T.hh = seconds / 3600 ; // do 1 giờ = 3600 giây
```

```
    int s = seconds % 3600 ; // số giây đã trôi qua tính từ đầu giờ
```

```
    T.mm = s / 60 ;
```

```
    T.ss = s % 60;
```

```
    return T ;
```

```
}
```

```
int GanNhat ( THOIDIEM T1, THOIDIEM T2, THOIDIEM T3 ) {
```

```
    int t12 = ThoiGian (T1, T2) ; // số giây từ giữa 2 th/d T1 & T2
```

```
    int t13 = ThoiGian (T1, T3) ; // số giây từ giữa 2 th/d T1 & T3
```

```
    int t23 = ThoiGian (T2, T3) ; // số giây từ giữa 2 th/d T2 & T3
```

```
    if (t12< t13 && t12<t23) return 12 ;
```

```
    if (t13< t12 && t13<t23) return 13 ;
```

```
    return 23;
```

```
}
```

*** CT minh họa việc dùng các hàm trên:**

```
int main ( ) {
```

```
    THOIDIEM T1 = {12,34,56} , T2 = {1,2,3} , T3 = {21,0,0} ;
```

```
    cout << « Xet cac th/d T1, T2, T3 la : »; XuatThoiDiem(T1); XuatThoiDiem(T2); XuatThoiDiem(T3);
```

```
    cout << "\n So giay tu 0 gio den T1 la : " << ThoiGian (T1) ;
```

```
    cout << "\n So giay giua 2 th/d T1 & T2 la : " << ThoiGian (T1,T2) ;
```

```
    int t = 2021 ;
```

```
    THOIDIEM T = ThoiDiem (t) ;
```

```
    cout<< "\n Thoi diem ung voi " << t << " giay (ke tu 0 giờ) la : " ; XuatThoiDiem(T);
```

```
    cout << "\n Voi 3 th/d T1, T2, T3 tren thi 2 th/d gan nhat la : " << GanNhat (T1, T2, T3) ;
```

```
}
```