

CHƯƠNG 2: CÁC KIỂU DỮ LIỆU CƠ SỞ

□ Bài tập có hướng dẫn

1. Nhập vào 2 số nguyên. Tính và in ra tổng, hiệu, tích, thương và dư của hai số nguyên này.

Hướng dẫn

```
#include <stdio.h>

void main() {
    int a, b;
    printf("Nhap hai so nguyen:");
    scanf("%d%d", &a, &b);
    printf("a + b = %d\n", a+b);
    printf("a - b = %d\n", a-b);
    printf("a * b = %d\n", a*b);
    printf("a / b = %d\n", a/b);
    printf("du = %d\n", a%b);
}
```

2. Nhập vào bán kính của hình tròn. Tính và in ra chu vi, diện tích theo công thức sau:

$$CV = 2 * PI * r$$

$$DT = PI * r * r$$

Hướng dẫn

```
#define PI 3.1416

float r, cv, dt;
printf("Nhap ban kinh:");
scanf("%f", &r);
cv = 2 * r * PI;
dt = PI * r * r;
printf("Chu vi: %0.2f\n", cv);
printf("Dien tich: %0.2f\n", dt);
```

3. Nhập vào độ Ferenheit. Tính và in ra độ celsius theo công thức sau:

$$C = 5/9 * (F - 32)$$

Hướng dẫn

```
float c, f;
<Nhap f>
```

```
c = 5.0 / 9 * (f - 32); /*Hoặc c = 5 / 9.0 * (f - 32) hoặc c = 5.0 / 9.0 * (f - 32)*/
<In c>
```

4. Nhập khối lượng hai vật thể và khoảng cách giữa chúng. Tính và in ra lực vạn vật hấp dẫn theo công thức:

$$F = G \cdot (m_1 \cdot m_2) / r^2$$

G: hằng số hấp dẫn ($6.67 \cdot 10^{-11} \text{ Nm}^2/\text{kg}^2$)

m_1, m_2 : Khối lượng hai vật thể.

r: khoảng cách hai vật.

Hướng dẫn

```
#define G 6.67E-11 //Cách viết khoa học của 6.67*10-11
```

5. Nhập vào một số nguyên n và một số thực x. Tính và in ra biểu thức $(x^2 + 1)^n$.

Hướng dẫn

```
#include <math.h>
float x, kq;
int n;
<Nhập n, x>
kq = pow(x * x + 1, n);
<In kq>
```

6. Nhập vào một số tiền nguyên dương. Đổi số tiền này ra các tờ giấy bạc 50đ, 20đ, 10đ, 5đ và 1đ. Với giả thiết ưu tiên cho tờ có mệnh giá lớn hơn, hãy in ra xem đổi được bao nhiêu tờ mỗi loại.

Hướng dẫn:

```
int tien;
int to50, to20, to10, to5, to1;
<Nhập tien>
to50 = tien / 50;
to20 = tien % 50 / 20;
...
<In to50, to20, to10, to5, to1>
```

☐ Bài tập luyện tập

7. Nhập vào chiều dài và chiều rộng của hình chữ nhật. Tính và in ra chu vi, diện tích theo công thức sau:

$$CV = (a+b) \cdot 2$$

$$DT = a \cdot b$$

8. Nhập vào 3 cạnh của hình tam giác. Tính và in ra chu vi, diện tích theo công thức sau:

$$CV = a+b+c$$

$$DT = (p*(p-a)*(p-b)*(p-c))^{1/2}$$

với p là nửa chu vi.

9. Nhập vào bán kính hình cầu. Tính và in ra thể tích, diện tích theo công thức sau:

$$TT = 4*PI*r^3 / 3;$$

$$DT = 4*PI*r^2$$

10. Nhập vào chiều dài con lắc đơn. Tính và in ra chu kỳ con lắc đơn theo công thức:

$$T = 2*PI*(l/g)^{1/2}$$

l: chiều dài con lắc đơn

g: gia tốc trọng trường (9.81 m/s²)

11. Nhập vào một số nguyên n và một số thực x. Tính và in ra biểu thức

$$A = (x^2 + x + 1)^n + (x^2 + x - 1)^n$$

12. Nhập vào một số nguyên dương có đúng 3 chữ số. Tính và in ra số đảo ngược.

Hướng dẫn

Từ số nhập vào hãy tính các chữ số ở hàng đơn vị, hàng chục và hàng trăm. Sau đó số đảo được tính theo công thức:

$$\text{Số đảo} = \text{đơn vị} * 100 + \text{chục} * 10 + \text{trăm}$$

CHƯƠNG 3: CÁC CÂU LỆNH Rẽ NHÁNH

□ Bài tập có hướng dẫn

1. Nhập vào một ký tự. Hãy in thông báo cho biết ký tự đó là ký tự số, ký tự chữ cái, ký tự phép toán hay là ký tự dạng khác với các dạng trên.

Hướng dẫn

```
char ch;
<Nhập ch>
if(ch >= '0' && ch <= '9')
    <Day la ky tu so>
else if((ch >= 'a' && ch <= 'z') || (ch >= 'A' && ch <= 'Z'))
    <Day la ky tu chu cai>
else if(ch == '+' || ch == '-' || ch == '*' || ch == '/' || ch == '%')
    <Day la ky tu phep toan>
else <Day la ky tu dang khac>
```

2. Nhập vào hai số nguyên. Hãy mô phỏng một máy tính đơn giản gồm 5 phép tính hai ngôi (+, -, *, /, %) trên hai số nguyên này.

Hướng dẫn

```
int a, b;
char pt;           //Phép toán
<Nhập a>
fflush(stdin);     //Xóa dữ liệu trong vùng đệm
<Nhập pt>
<Nhập b>
switch(pt) {
    case '+': <In a + b>; break;
    case '-': <In a - b>; break;
    case '*': <In a * b>; break;
    case '/': <In a / b>; break;
    case '%': <In a % b>; break;
    default: <Phep toan khong hop le>
}
```

3. Nhập vào hai số thực a, b. Hãy tính và in ra nghiệm phương trình bậc nhất:

$$ax + b = 0.$$

Hướng dẫn

```
float a, b;
float x; //Nghiem
<Nhap a, b>
if(a == 0) {
    if(b == 0)
        <Phuong trinh vo so nghiem>
    else <Phuong trinh vo nghiem>
}
else {
    x = -b / a;
    <In nghiem x>
}
```

4. Nhập vào ba số thực a, b, c (a khác 0). Hãy tính và in ra nghiệm phương trình bậc hai:

$$ax^2 + bx + c = 0.$$

Hướng dẫn

```
float a, b, c;    //Lưu ba hệ số
float x1, x2;    //Lưu nghiệm
float delta;      //Lưu biệt số delta
<Nhap a, b, c>
delta = b * b - 4 * a * c;
if(delta < 0)
    <Phuong trinh vo nghiem>
else if(delta == 0) {
    x1 = x2 = -b / (2 * a);
    <In x1 hoac x2>
}
else {
    x1 = (-b + sqrt(delta)) / (2 * a);
    x2 = (-b - sqrt(delta)) / (2 * a);
    <In x1 va x2>
}
```

5. Nhiệt độ F(Fahrenheit) và nhiệt độ C(Celcius) liên hệ với nhau theo công thức: $C = 5(F - 32)/9$. Viết chương trình cho phép người dùng nhập vào độ F hay độ C và đổi sang độ còn lại.

Hướng dẫn

```
float f, c;
int chon;      //Chức năng chọn
printf("BANG CHON\n");
printf("1: Nhap F, tinh C\n");
printf("2: Nhap C, tinh F\n");
printf("Moi chon:"); scanf("%d", &chon);
switch(chon) {
case 1:
    <Nhap f>;
    <Tinh c>;
    <In c>;
    break;
case 2:
    <Nhap c>;
    <Tinh f>;
    <In f>;
    break;
default: <Ban chon khong hop le>
}
```

6. Nhập vào ngày, tháng, năm. Kiểm tra xem ngày, tháng, năm đó có hợp lệ hay không, nếu hợp lệ hãy cho biết ngày hôm sau của ngày đã nhập là ngày nào.

(Biết rằng năm nhuận là năm (chia hết cho 400) hoặc (chia hết cho 4 và không chia hết cho 100)).

Ví dụ: Nhập 31-12-2000 → Ngày hôm sau là 01-01-2001

Hướng dẫn

```
int nhuan; //Biến cờ = 0: Không là năm nhuận, = 1: Là năm nhuận
int d, m, y;      //Ngày nhập
int nd, nm, ny; //Ngày kế tiếp
int snnt;         //Số ngày trong tháng
<Nhap d, m, y>
//Kiểm tra năm nhuận
nhuan = 0;
if(y % 400 == 0 || (y%4 == 0 && y % 100 != 0))
    nhuan = 1;
//Tính số ngày trong tháng
switch(m) {
case 1: case 3: case 5: case 7: case 8: case 10: case 12:
```

```

        snnt = 31; break;
    case 4: case 6: case 9: case 11:
        snnt = 30; break;
    case 2:
        snnt = (nhuan ? 29 : 28); break;
    default: snnt = 0;
}
//Kiểm tra hợp lệ
if(d > 0 && d <= snnt && m > 0 && m < 13 && y > 0) {
    //Tính ngày kế tiếp
    nd = d + 1; nm = m; ny = y;
    if(nd > snnt) {
        nd = 1;
        nm++;
        if(nm > 12) {
            nm = 1;
            ny++;
        }
    }
    <In nd, nm, ny>
}
else <Ngày nhập không hợp lệ>

```

7. Nhập vào giờ, phút, giây. Kiểm tra giờ, phút giây đó có hợp lệ hay không, nếu hợp lệ hãy cho biết giờ sau đó 1 giây là bao nhiêu.

Ví dụ: Nhập 01:59:59 → Giờ sau đó 1 giây là 02:00:00

Hướng dẫn

```

<Nhập h, m, s>
if(h >= 0 && h < 24 && m >= 0 && m < 59 && s >= 0 && s < 59) {
    ns = s + 1 ; nm = m ; nh = h ;
    if(ns > 59) {
        ns = 0
        nm++ ;
        if(nm > 59) {
            nm = 0 ;
            nh++ ;

```

```

    }
}
else <Gio nhap khong hop le>

```

8. Nhập vào số ngày thuê và loại phòng (một trong 3 loại A, B hoặc C). Hãy tính và in ra tiền thuê phòng với quy định như sau:

Loại A: 250000đ/ngày

Loại B: 200000đ/ngày

Loại C: 150000đ/ngày

Nếu thuê quá 12 ngày thì phần trăm được giảm trên tổng số tiền (tính theo giá quy định) là: 10% cho phòng loại A, 8% cho phòng loại B hoặc C.

Hướng dẫn

```

#define DON_GIA_A    250000.0
#define DON_GIA_B    200000.0
#define DON_GIA_C    150000.0
#define GIAM_A        0.1
#define GIAM_B_C      0.08
#define GIOI_HAN      12

int snt;           //Số ngày thuê
char lp;           //Loại phòng
float tien;        //Tiền thuê phòng

<Nhập snt>
fflush(stdin); //Xóa dữ liệu trong vùng đệm
<Nhập lp>
if(snt > 0) {
    switch(lp) {
        case 'a': case 'A':
            tien = DON_GIA_A * snt;
            if(snt > GIOI_HAN)
                tien *= (1 - GIAM_A);
            <In tien>
            break;
        case 'b': case 'B': ...
        case 'c': case 'C': ...
        default: <Loai phong khong hop le>
    }
}

```



```
}
else <Số ngày thuê không hợp lệ>
```

□ Bài tập luyện tập

9. Nhập vào ba số thực a, b, c (a bất kỳ). Hãy tính và in ra nghiệm phương trình bậc hai $ax^2 + bx + c = 0$.

10. Nhập vào sáu số thực a, b, c, d, m, n . Hãy tính và in ra nghiệm hệ phương trình hai ẩn số:

$$ax + by = m$$

$$cx + dy = n$$

Nhắc lại:

Tính $DX = md - nb$, $DY = an - cm$ và $D = ad - cb$.

Nghiệm: $x = DX/D$, $y = DY/D$ (D khác 0)

11. Nhập vào các số thực a, b, c . Hãy kiểm tra xem ba số này có lập thành 3 cạnh của một tam giác hay không? Nếu có hãy tính và in ra chu vi, diện tích, chiều dài mỗi đường cao của tam giác.

12. Nhập vào một số nguyên gồm 3 chữ số. Xuất ra màn hình chữ số lớn nhất ở vị trí nào.

Ví dụ: Nhập vào 384 → Chữ số lớn nhất nằm ở hàng chục

13. Nhập vào ba số nguyên d, m, y là ngày, tháng, năm. Hãy in thông báo cho biết đó là ngày thứ mấy trong tuần. Sử dụng công thức sau để biến đổi ngày, tháng, năm thành thứ trong tuần:

$$A = d + 2m + (3(m + 1)/5) + y + (y/4) - (y/100) + (y/400) + 2$$

với quy ước tháng 1, 2 của năm y được xem là tháng 13, 14 của năm $y-1$. Số dư trong phép chia A cho 7 cho kết quả là thứ trong tuần theo nghĩa số dư là 0: Thứ bảy, số dư là 1: Chủ nhật, số dư là 2: Thứ hai. v.v...

14. Nhập vào ngày, tháng, năm. Kiểm tra xem ngày, tháng, năm đó có hợp lệ không, nếu hợp lệ hãy cho biết ngày hôm trước của ngày đã nhập là ngày nào.

(Biết rằng năm nhuận là năm (chia hết cho 400) hoặc (chia hết cho 4 và không chia hết cho 100)).

Ví dụ: Nhập 01-01-2001 → Ngày hôm sau là 31-12-2000

15. Nhập vào giờ, phút, giây. Kiểm tra giờ, phút giây đó có hợp lệ hay không, nếu hợp lệ hãy cho biết giờ trước đó 1 giây là bao nhiêu.

Ví dụ: Nhập 02:00:00 → Giờ sau đó 1 giây là 02:00:01

16. Nhập vào giờ vào ca, giờ ra ca. Hãy tính và in ra tiền lương ngày cho công nhân, biết rằng tiền trả cho mỗi giờ trước 12 giờ trưa là 6000đ và mỗi giờ sau 12 giờ trưa là 7500đ. Giờ vào ca sớm nhất là 6 giờ sáng và giờ ra ca trễ nhất là 18 giờ.

17. Nhập vào trọng lượng m (kg) hàng hóa bán được ($0 < m \leq 100$). Hãy tính và in ra tiền lời thu được, biết rằng

$0 < m \leq 10$ Tiền lời là 5000đ/kg.

$10 < m \leq 20$ Tiền lời là 5000đ/kg.

$20 < m \leq 50$ Tiền lời là 9000đ/kg và thêm 2% tổng số tiền lời.

$50 < m$ Tiền lời là 10000đ/kg và thêm 4% tổng số tiền lời nhưng không được quá 1000000đ.

18. Trong một kỳ thi tuyển, hội đồng tuyển sinh đã xem xét đề nghị một điểm chuẩn (xem như dữ kiện nhập). Mỗi thí sinh tham gia kỳ thi sẽ trúng tuyển nếu điểm tổng kết của thí sinh đó lớn hơn hoặc bằng điểm chuẩn và không có môn nào điểm 0. Điểm tổng kết của mỗi thí sinh là tổng điểm của 3 môn thi và điểm ưu tiên. Điểm ưu tiên bao gồm điểm ưu tiên theo khu vực và điểm ưu tiên theo đối tượng dự thi. Cho biết:

Khu vực A: điểm ưu tiên là 2.

Khu vực B: điểm ưu tiên là 1.

Khu vực C: điểm ưu tiên là 0.5.

Đối tượng 1: điểm ưu tiên là 2.5.

Đối tượng 2: điểm ưu tiên là 1.5.

Đối tượng 3: điểm ưu tiên là 1.

Viết chương trình nhập vào điểm chuẩn, điểm của 3 môn thi của một thí sinh, khu vực và đối tượng dự thi.

Cho biết thí sinh đó đậu hay rớt.

CHƯƠNG 4: CÁC CÂU LỆNH LẶP

□ Bài tập có hướng dẫn

1. Nhập vào số nguyên dương n . Hãy tính và in ra các tổng sau:

a. $s = 1 \cdot 2 + 2 \cdot 3 + \dots + n \cdot (n + 1)$

b. $s = 1^1 + 2^2 + \dots + n^n$

c. $s = 1 + 1/2 + 1/3 + \dots + 1/n$

Hướng dẫn

```
//Câu 1a
s = 0;
for(i = 1; i <= n; i++)
    s += i * (i + 1);

//Câu 1b
s = 0;
for(i = 1; i <= n; i++)
    s += pow(i, i);

//Câu 1c
s = 0.0; //s kiểu thực
for(i = 1; i <= n; i++)
    s += 1.0/i;
```

2. Nhập vào một số nguyên dương n . Hãy tính và in ra giá trị của vế trái, vế phải của mỗi công thức sau đây và cho biết kết quả tính của hai vế có bằng nhau hay không?

a. $1 \cdot 2 + 2 \cdot 2 + \dots + n \cdot 2 = n(n + 1)(2n + 1) / 6$

b. $1/2! + 2/3! + \dots + n/(n + 1)! = 1 - (1/(n + 1)!)$

Hướng dẫn

```
//Câu 2a
//Tính vế trái
vt = 0;
for(i = 1; i <= n; i++)
    vt += i * 2;

//Tính vế phải
vp = n * (n + 1) * (2 * n + 1) / 6;

<ln vt, vp>;

//So sánh hai vế
```

```

if(vt == vp)
    <Hai ve bang nhau>
else <Hai ve khong bang nhau>
//Câu 2b
#define E      0.0001 //Độ chính xác
    //Tính vế trái
vt = 0;
gt = 1;
for(i = 1; i <= n; i++) {
    gt *= (i + 1);
    vt += (float)i/gt;
}
    //Tính vế phải
vp = 1 - (1.0/gt);
    //So sánh hai vế
if(fabs(vt - vp) < E)
    <Hai ve bang nhau>
else <Hai ve khong bang nhau>
    
```

Lưu ý: Các phép toán trên số thực trong máy tính đều có những sai số nhất định. Do đó khi so sánh hai giá trị thực a và b nên dùng biểu thức $|a - b| < E$, trong đó E gọi là độ sai số cho phép (còn gọi là độ chính xác).

3. Nhập vào x (theo radian). Hãy tính $\sin(x)$ theo công thức sau:

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots + (-1)^n \frac{x^{2n+1}}{(2n+1)!} \quad (\text{độ chính xác là } 0.0001)$$

Lưu ý: Tính $\sin(x)$ với độ chính xác là 0.0001 có nghĩa là $\sin(x)$ được tính đến số hạng thứ n nào đó sao cho trị tuyệt đối của số hạng đó nhỏ hơn 0.0001.

Hướng dẫn

```

#define E      0.0001      //độ chính xác
<Nhập x>
k = 1; y = x; sin = x;
while(fabs(y) >= E) {
    k += 2;
    y = -y * x * x / ((k - 1) * k);
    sin += y;
}
    
```

<In sin>

4. Hãy in bảng mã ASCII ra màn hình. Bảng được in ra theo từng trang màn hình sao cho người sử dụng có thể đọc được.

Hướng dẫn:

```
#define NUM_LINES 40      //Giả sử một trang màn hình có 40 dòng
for(i = 0; i < 256; i++) {
    printf("%c: %d\n", i, i);
    if((i + 1) % NUM_LINES == 0) {
        printf("*** Nhan phim bat ky de tiep tuc **\n");
        getch();          //Dừng màn hình
    }
}
```

5. Nhập vào một số nguyên. Hãy kiểm tra xem số nguyên đó có phải là số nguyên tố hay không? Biết rằng một số nguyên dương (khác số 1) được gọi là số nguyên tố nếu ngoài 1 và chính nó ra thì nó không chia hết cho một số nguyên dương nào khác. Chẳng hạn các số sau là nguyên tố:

2, 3, 5, 7, 11, 13, . . .

Hướng dẫn:

```
int n;
int nt;      //0: Không là nguyên tố, 1: Là nguyên tố (cờ hiệu)
<Nhập n>
if(n < 2)
    nt = 0;      //Không là số nguyên tố
else {
    nt = 1;      //Giả sử là số nguyên tố
    for(i = 2; i < n; i++) {
        if(n % i == 0) {
            nt = 0;      //Không là số nguyên tố
            break;
        }
    }
}
//In kết quả
if(nt) <Day la so nguyen to>
else <Day khong la so nguyen to>
```

6. Nhập vào một số nguyên. Hãy kiểm tra xem một số nguyên đó có phải là số hoàn chỉnh hay không? Biết rằng một số nguyên dương n được gọi là số hoàn chỉnh nếu tổng các ước số của n (không kể n) bằng chính nó. Chẳng hạn các số sau là hoàn chỉnh:

6, 28, 496, ...

Hướng dẫn:

```
int n;
int hc; //0: Không là số hoàn chỉnh, 1: Là số hoàn chỉnh (cờ hiệu)
<Nhập n>
if(n < 1)
    hc = 0; //Không là số hoàn chỉnh
else {
    //Tính tổng các ước số của n
    s = 0;
    for(i = 1; i < n; i++)
        if(n % i == 0) s += i;
    //Kiểm tra
    hc = (s == n ? 1 : 0);
}
//In kết quả
if(hc)
    <Đây là số hoàn chỉnh>
else <đây không là số hoàn chỉnh>
```

7. Nhập vào một số nguyên. Hãy kiểm tra xem số nguyên đó có phải là số chính phương hay không? Biết rằng một số nguyên dương n được gọi là số chính phương nếu nó bằng bình phương của một số nguyên. Chẳng hạn các số sau là chính phương:

1, 4, 9, 16, 25, ...

Hướng dẫn:

```
int n;
int cp; //0: Không là số cphương, 1: Là số cphương (cờ hiệu)
<Nhập n>
if(n < 1)
    cp = 0; // Không là số chính phương
else {
    cp = 0; //Giả sử không là số chính phương
    for(i = 1; i <= sqrt(n); i++) {
        if(n == i * i) {
```

```

                                cp = 1;          //Là số chính phương
                                break;

                                }

                                }

                                }

//In kết quả
if(cp) <Day la so chinh phuong >
else <Day khong la so chinh phuong >

```

8. Nhập vào một số nguyên. Hãy kiểm tra xem số nguyên đó có phải là số đối xứng hay không? Biết rằng một số nguyên n được gọi là đối xứng nếu số đảo của n bằng chính n. Chẳng hạn các số sau là đối xứng:

1, 44, 161, 2552, ...

Hướng dẫn:

```

int n;
int m; //Lưu lại giá trị của n
int dao; //Số đảo
int dx; //0: Không là số đối xứng, 1: Là số đối xứng (cờ hiệu)
int d;

<nhap n>

m = n;
//Tìm số đảo của m
dao = 0;
while(m > 0) {
    d = m % 10;          // Lấy chữ số phải nhất của m
    dao = 10*dao + d;      // Thêm chữ số phải nhất vào dao
    m /= 10;              // Xóa chữ số phải nhất của m
}
dx = (dao == n ? 1: 0);
//Kiểm tra đối xứng
if(dx == 1)
    <Day la so doi xung>
else <Day la so khong doi xung>

```

9. Nhập vào hai số nguyên a và b. Hãy tìm và in ra ước số chung lớn nhất của a và b. Áp dụng giải thuật **Euclide** sau (chỉ áp dụng cho hai số nguyên dương):

- (1) Lấy số lớn trừ số nhỏ
- (2) Đặt số lớn bằng hiệu số của phép trừ ở bước (1)
- (3) Lặp lại bước (1) cho đến khi hai số bằng nhau

Giá trị bằng nhau đạt được chính là ước số chung lớn nhất của hai số đã cho.

Hướng dẫn:

```
int a, b;
int uscln;      //Ước số chung lớn nhất
<Nhập a, b>
if(a < 0)      a = -a;
if(b < 0)      b = - b;
if(a == 0 || b == 0)
    uscln = a + b;
else {
    while(a != b) {
        if(a > b) a = a - b;
        else b = b - a;
    }
    uscln = a; //hoặc b
}
<In uscln>
```

❑ Bài tập luyện tập

10. Nhập vào một số nguyên dương n. Hãy tính và in ra giá trị của vế trái, vế phải của mỗi công thức sau đây và cho biết kết quả tính của hai vế có bằng nhau hay không?
 - a. $1.2.3 + 2.3.4 + \dots + n(n+1)(n+2) = n(n+1)(n+2)(n+3)/4$
 - b. $1(1!) + 2(2!) + \dots + n(n!) = (n+1)! - 1$
11. Nhập vào x (tính theo radian). Hãy tính e(x) và cos(x) theo các công thức sau:

$$e(x) = 1 + x + x^2/2! + \dots + x^n/n! \text{ (độ chính xác là 0.0001)}$$

$$\cos(x) = 1 - x^2/2! + x^4/4! - \dots + (-1)^n(x^n/2n!) \text{ (độ chính xác là 0.0001)}$$
12. Hãy in toàn bộ bảng cửu chương từ hai đến chín ra màn hình. Bảng được in ra theo từng trang màn hình sao cho người sử dụng có thể đọc được.
13. Nhập vào số nguyên dương n. Hãy tìm và in ra các số nguyên tố trong khoảng từ 1 đến n.
14. Nhập vào số nguyên dương n. Hãy tìm và in ra các số hoàn chỉnh trong khoảng từ 1 đến n.
15. Nhập vào số nguyên dương n. Hãy tìm và in ra các số chính phương trong khoảng từ 1 đến n.
16. Nhập vào số nguyên dương n. Hãy tìm và in ra các số đối xứng trong khoảng từ 1 đến n.

CHƯƠNG 5: HÀM

❑ Bài tập có hướng dẫn

1. Viết hàm `DienTich()` sau, hàm này trả về diện tích một đường tròn với bán kính `r`:

`float DienTich (float r);`

Hướng dẫn

```
float DienTich (float r) {
    return r * r * PI;
}
```

2. Viết hàm `Min()` sau, hàm này trả về giá trị nhỏ nhất của hai số nguyên `x` và `y`:

`int Min(int x, int y);`

Hướng dẫn :

```
int Min(int x, int y) {
    return (x > y ? x : y) ;
}
```

3. Viết hàm `LuyThua()` sau, hàm này trả về giá trị x^p :

`float LuyThua (float x, int p);`

Hướng dẫn:

```
float LuyThua(float x, int p) {
    int i ;
    float ret = 1 ;
    for(i = 1 ; i <= p ; i++)
        ret *= x ;
    return ret ;
}
```

4. Viết hàm `ToHop()` sau, hàm này trả về $C(n, k)$ là số tổ hợp n lấy k ($n > 0, 0 \leq k \leq n$) bằng cách dựa vào công thức $C(n, k) = n! / (k!(n - k)!)$:

`long ToHop (int n, int k);`

Hướng dẫn:

```
long GiaiThua(int u) {
    long ret = 1;
    int i;
    for(i = 1; i <= u; i++)
```

```

        ret *= i;
    return ret;
}
long ToHop(int n, int k) {
    return GiaiThua(n)/(GiaiThua(k)*GiaiThua(n-k));
}

```

5. Viết hàm Nhuận() sau, hàm này trả về giá trị 1 (đúng) nếu năm y nhuận và trả về giá trị 0 (sai) nếu năm y không nhuận:

int Nhuận (int y);

Hướng dẫn

```

int Nhuận(int y) {
    if(y % 400 == 0 || (y % 4 == 0 && y % 100 != 0))
        return 1 ;
    else return 0 ;
}

```

6. Viết hàm SNTrongThang () sau, hàm này trả về số ngày trong tháng m và năm y:

int SNTrongThang (int m, int y);

Hướng dẫn :

```

int SNTrongThang (int m, int y) {
    int ret = 0 ;
    switch(m) {
    case 1 : case 3 : case 5 : case 7 : case 8 : case 12 :
        ret = 31; break ;
    case 4 : case 6 : case 9 : case 11 :
        ret = 30; break ;
    case 2:
        ret = (Nhuận(y) ? 29 : 28) ; break ;
    }
}

```

7. Viết hàm NguyênTo() sau, hàm này trả về giá trị 1 (đúng) nếu x là số nguyên tố và trả về giá trị 0 (sai) nếu x không là số nguyên tố:

int NguyênTo (int x);

Hướng dẫn

```

int NguyênTo(int x) {

```

```

int nt;          //biến cờ = 1: x là nguyên tố, 0: x không là nguyên tố
int i;
if(x < 2)
    nt = 0;      //Không là nguyên tố
else {
    nt = 1;      //Là nguyên tố
    for(i = 2; i < x; i++)
        if(x % i == 0) {
            nt = 0;      //Không là nguyên tố
            break;
        }
    }
return nt;
}

```

8. Viết hàm HoanChinh() sau, hàm này trả về giá trị 1 (đúng) nếu x là số hoàn chỉnh và trả về giá trị 0 (sai) nếu x không là số hoàn chỉnh:

int HoanChinh (int x);

Hướng dẫn:

```

int HoanChinh(int x) {
    int hc;          //Biến cờ
    int s;
    if(x <= 0)
        hc = 0;
    else {
        s = 0;
        for(i = 1; i < x; i++)
            if(x % i == 0) s += i;
        hc = (s == x ? 1: 0);
    }
    return hc;
}

```

9. Viết hàm ChinhPhuong() sau, hàm này trả về giá trị 1 (đúng) nếu x là số chính phương và trả về giá trị 0 (sai) nếu x không là số chính phương:

int ChinhPhuong (int x);

Hướng dẫn:

```

int ChinhPhuong(int x) {

```

```

int cp; //0: Không là số cphương, 1: Là số cphương (cờ hiệu)
if(x < 1)
    cp = 0;          // Không là số chính phương
else {
    cp = 0;          //Giả sử không là số chính phương
    for(i = 1; i <= sqrt(x); i++) {
        if(x == i * i) {
            cp = 1;      //Là số chính phương
            break;
        }
    }
}
return cp;
}

```

10. Viết hàm DoiXung() sau, hàm này trả về giá trị 1 (đúng) nếu x là số đối xứng và trả về giá trị 0 (sai) nếu x không là số đối xứng:

int DoiXung (int x);

Hướng dẫn:

```

int DoiXung(int x) {
    int dx;          //Biến Cờ
    int dao;         //Lưu số đảo
    int y;
    y = x;
    //Tìm số đảo của y
    dao = 0;
    while(y > 0) {
        d = y % 10;      // Lấy chữ số phải nhất của y
        dao = 10*dao + d; // Thêm chữ số phải nhất vào dao
        y /= 10;         // Xóa chữ số phải nhất của y
    }
    dx = (dao == x ? 1: 0);
    return dx ;
}

```

11. Viết hàm TimUCLN() sau, hàm này trả về ước số chung lớn nhất của hai số nguyên a và b:

int TimUCLN (int a, int b);

Hướng dẫn:

```
int TimUCLN (int a, int b) {
    int ret;
    if(a < 0)      a = -a;
    if(b < 0)      b = - b;
    if(a == 0 || b == 0)
        ret = a + b;
    else {
        while(a != b) {
            if(a > b) a = a - b;
            else b = b - a;
        }
        uscln = a; //hoặc b
    }
}
```

12. Viết hàm RutGon() sau, hàm này rút gọn một phân số gồm ts và ms:

```
void RutGon (int *ts, int *ms);
```

Hướng dẫn:

```
void RutGon (int *ts, int *ms) {
    int uscln = TimUCLN(*ts, *ms);
    *ts /= uscln;
    *ms /= uscln;
}
```

13. Viết hàm TinhDTr() sau, hàm này trả về diện tích dt và chu vi cv của một đường tròn bán kính r:

```
void TinhDTr (float r, float *dt, float *cv);
```

Hướng dẫn:

```
void TinhDTr (float r, float *dt, float *cv) {
    *dt = r * r * PI;
    *cv = 2 * r * PI;
}
```

14. Viết hàm GiaiPTBac2() sau, hàm này trả về số nghiệm sn và các nghiệm x1, x2 (nếu có) của một phương trình bậc 2 với các hệ số a, b và c:

```
void GiaiPTBac2 (float a, float b, float c, int *sn, float *x1, float *x2);
```

Hướng dẫn:

```
void GiaiPTBac2 (float a, float b, float c, int *sn, float *x1, float *x2) {
    float delta = b*b - 4*a*c ;
```

```

if(delta < 0)
    *sn = 0
else if(delta == 0) {
    *sn = 1;
    *x1 = *x2 = -b / (2*a);
}
else {
    *sn = 2;
    *x1 = (-b + sqrt(delta)) / (2*a);
    *x2 = (-b - sqrt(delta)) / (2*a);
}
}

```

15. Viết hàm NgayKe() sau, hàm này trả về ngày kế tiếp nd, nm, ny của ngày hiện hành d, m, y:

```
void NgayKe(int d, int m, int y, int *nd, int *nm, int *ny);
```

Hướng dẫn:

```

void NgayKe(int d, int m, int y, int *nd, int *nm, int *ny) {
    *nd = d + 1, *nm = m; *ny = y;
    if(*nd > SNTrongThang(*nm, *ny)) {
        *nd = 1;
        (*nm)++;
        if(*nm > 12) {
            *nm = 1;
            (*ny)++;
        }
    }
}
}

```

❑ Bài tập luyện tập

16. Viết hàm Min() sau, hàm này trả về giá trị nhỏ nhất của ba số nguyên x, y và z:

```
int Min(int x, int y, int z);
```

17. Viết hàm InTGPascal() sau, hàm này xuất ra một tam giác Pascal có sd dòng. Hình sau đây là một tam giác Pascal có 7 dòng

	0	1	2	3	4	5	6
0	1						
1	1	1					

2	1	2	1				
3	1	3	3	1			
4	1	4	6	4	1		
5	1	5	10	10	5	1	
6	1	6	15	20	15	6	1

Mỗi số trong tam giác Pascal là một tổ hợp $C(n, k)$. Nếu chúng ta đếm dòng và cột bắt đầu bằng 0, thì một số ở dòng n và cột k là $C(n, k)$. Ví dụ, số ở dòng 6 và cột 4 là $C(6, 4) = 15$
 void InTGPascal(int sd);

18. Viết hàm HopLe() sau, hàm này trả về giá trị 1 (đúng) nếu ngày d , tháng m , năm y hợp lệ và trả về giá trị 0 (sai) nếu d , m , y không hợp lệ:

int HopLe(int d, int m, int y);

19. Viết hàm InNguyenTo() sau, hàm này in các số nguyên tố trong khoảng từ số đầu đến số cuối:

void InNguyenTo(int dau, int cuoi);

20. Viết hàm InHoanChinh() sau, hàm này in các số hoàn chỉnh trong khoảng từ số đầu đến số cuối:

void InHoanChinh (int dau, int cuoi);

21. Viết hàm InChinhPhuong () sau, hàm này in các số chính phương trong khoảng từ số đầu đến số cuối:

void InChinhPhuong (int dau, int cuoi);

22. Viết hàm InDoiXung () sau, hàm này in các số đối xứng trong khoảng từ số đầu đến số cuối

void InDoiXung (int dau, int cuoi);

23. Viết hàm TinhHCN() sau, hàm này trả về diện tích dt và chu vi cv của một hình chữ nhật với chiều dài là d và chiều rộng là r :

void TinhHCN (float d, float r, float *dt, float *cv);

24. Viết hàm TinhTG() sau, hàm này trả về diện tích dt và chu vi cv của một tam giác với các cạnh a , b và c :

void TinhTG (float a, float b, float c, float *dt, float *cv);

25. Viết hàm GiaiPTTrungPhuong()sau, hàm này trả về số nghiệm sn và các nghiệm x_1, x_2, x_3, x_4 (nếu có) của một phương trình trùng phương với các hệ số a, b và c :

void GiaiPTTrungPhuong (float a, float b, float c, int *sn, float *x1, float *x2, float *x3, float *x4);

26. Viết hàm NgayTruoc() sau, hàm này trả về ngày kế trước pd, pm, py của ngày hiện hành d, m, y :

void NgayTruoc(int d, int m, int y, int *pd, int *pm, int *py);

CHƯƠNG 6: MẢNG MỘT CHIỀU

Đối với các bài tập trong chương này, trong chương trình luôn phải có hai hàm nhập và xuất mảng một chiều nguyên.

❑ Bài tập có hướng dẫn

1. Viết chương trình thực hiện các chức năng sau:

- Tính tổng các phần tử có chữ số đầu tiên là chữ số lẻ.
- Tính tổng các phần tử là số chính phương.

Hướng dẫn:

```
//Câu 1a
int ChuSoDauTienLe(int x) {
    while (x >= 10)
        x /= 10 ;
    return (x % 2 != 0 ? 1 : 0) ;
}

int TongChuSoDauTienLe(int a[], int n) {
    int ret = 0;
    for(i = 0; i < n; i++)
        if(ChuSoDauTienLe(a[i]))
            ret += a[i];

    return ret;
}

//Câu 1b
int TongChinhPhuong(int a[], int n) {
    int ret = 0;
    for(i = 0; i < n; i++)
        if(ChinhPhuong(a[i]))
            ret += a[i];

    return ret;
}
```

2. Viết chương trình thực hiện các chức năng sau:

- Đếm số phần tử có chữ số đầu tiên là chữ số lẻ.
- Đếm số phần tử là số chính phương.
- Đếm số phần tử phân biệt, các phần tử trùng nhau chỉ đếm một lần.

Hướng dẫn:

```
//Câu 2a
int DemChuSoDauTienLe(int a[], int n) {
    int ret = 0;
    for(i = 0; i < n; i++)
        if(ChuSoDauTienLe(a[i]))
            ret++;
    return ret;
}

//Câu 2b
int DemChinhPhuong(int a[], int n) {
    int ret = 0;
    for(i = 0; i < n; i++)
        if(ChinhPhuong(a[i]))
            ret++;
    return ret;
}

//Câu 2c
int DemPhanBiet(int a[], int n) {
    int ret = 0;
    for(i = 0; i < n; i++) {
        if(TimKiem(a, i, a[i]) == -1) /*Không tìm thấy a[i] trong
            mảng a gồm i phần tử */
            ret++;
    }
    return ret;
}
```

3. Viết chương trình thực hiện các chức năng sau:

- Kiểm tra mảng có chứa số dương hay không?
- Kiểm tra mảng có đối xứng hay không?
- Kiểm tra mảng có tất cả các cặp phần tử đứng cạnh nhau có giá trị khác nhau hay không?

Hướng dẫn:

```
//Câu 3a
int CoDuong(int a[], int n) {
    int ret = 0;           //Mảng không có số dương
    int i;
```

```

    for(i = 0; i < n - 1; i++)
        if(a[i] > 0) {
            ret = 1;          // Mảng có số dương
            break;
        }
    return ret;
}

//Câu 3b
int DoiXung(int a[], int n) {
    int ret = 1; //Mảng đối xứng
    int i;
    for(i = 0; i < n/2; i++)
        if(a[i] != a[n - 1 - i]) {          //Mảng không đối xứng
            ret = 0;
            break;
        }
    return ret;
}

//Câu 3c
int CapKhacNhau(int a[], int n) {
    int ret = 1;          //Mảng có cặp khác nhau
    int i;
    for(i = 0; i < n - 1; i++)
        if(a[i] == a[i + 1]) {
            ret = 0;          //Mảng có cặp không khác nhau
            break;
        }
    return ret;
}

```

4. Viết chương trình thực hiện các chức năng sau:

- Sắp xếp các phần tử là số chính phương tăng dần, các phần tử khác giữ nguyên vị trí.
- Sắp xếp nửa mảng đầu tăng dần, nửa mảng sau giảm dần.

Hướng dẫn:

```

//Câu 4a
void SXCPtang(int a[], int n) {

```

```

int i, j;
for(i = 0; i < n-1; i++)
    for(j = i+1; j < n; j++)
        if(ChinhPhuong(a[i]) && ChinhPhuong(a[j]) && a[i] > a[j]) {
            int tam = a[i];
            a[i] = a[j];
            a[j] = tam;
        }
}
//Câu 4b
void SXNuaTangNuaGiam(int a[], int n) {
    SapXepDayConTang(a, n, 0, n/2);
    SapXepDayConGiam(a[], n, n/2 + 1, n - 1)
}

```

5. Viết chương trình thực hiện các chức năng sau:

- a. Tìm vị trí phần tử nguyên tố nhỏ nhất.
- b. Tìm tất cả các cặp phần tử có tổng là số chính phương.

Hướng dẫn:

```

//Câu 6a
int TimVTNTMin(int a[], int n) {
    int i, j, ret;
    //Tìm vị trí phần tử nguyên tố đầu tiên
    for(i = 0; i < n; i++)
        if(NguyenTo(a[i])) break;
    if(i >= n) //Không có phần tử nguyên tố
        ret = -1;
    else { //Có phần tử nguyên tố
        //Tìm vị trí nguyên tố min
        ret = i;
        for(j = i + 1; j < n; j++)
            if(NguyenTo(a[j]) && a[j] < a[ret])
                ret = j;
    }
    return ret;
}

```

```
//Câu 6b
void TimCapCoTongLaCP(int a[], int n) {
    int qh = 0;
    int i, j;
    for(i = 0; i < n - 1; i++)
        for(j = i + 1; j < n; j++)
            if(ChinhPhuong(a[i] + a[j])) {
                printf("(%d, %d)\n", a[i], a[j]);
                qh = 1;
            }
    if(qh == 0)
        printf("Khong tim thay cap phan tu nao co quan he\n");
}
```

6. Viết chương trình thêm một giá trị x vào trong mảng có thứ tự tăng sao cho vẫn giữ nguyên tính tăng của mảng.

Hướng dẫn:

```
void ThemVaoMangTang(int a[], int *n, int x) {
    int i;
    //Tìm vị trí phần tử đầu tiên lớn hơn hoặc bằng x
    for(i = 0; i < n; i++)
        if(a[i] >= x) break;
    //Thêm x vào vị trí i
    Them(a, &(*n), i, x);
}
```

7. Viết các hàm thực hiện các chức năng sau:
- Xóa phần tử là số chính phương đầu tiên.
 - Xóa tất cả phần tử là số chính phương.

Hướng dẫn:

```
//Câu 7a
int TimCPDauTien(int a[], int n) {
    int ret = -1;
    int i;
    for(i = 0; i < n; i++)
        if(ChinhPhuong(a[i])) {
```

```

        ret = i;
        break;
    }
    return ret;
}

void XoaCPDauTien(int a[], int *n) {
    int vt = TimCPDauTien(a, *n);
    if(vt != -1) {
        Xoa(a, &(*n), vt);
    }
    else <Khong tim thay so CP>
}

//Câu 7b
void XoaTatCaCP(int a[], int *n) {
    int vt;
    while((vt = TimCP(a, *n)) != -1) {
        Xoa(a, &(*n), vt);
    }
}

```

8. Viết các hàm thực hiện các chức năng sau:

- Tạo mảng b từ mảng a sao cho mảng b chỉ chứa các phần tử phân biệt.
- Tạo mảng b từ mảng a sao cho mảng b chỉ chứa các phần tử là số chính phương.

Hướng dẫn:

```

//Câu 8a
void TaoMangChuaPhanBiet(int a[], int n, int b[], int *m) {
    *m = 0;
    for(int i = n; i > 0; i--)
        if(TimKiem(a, i, a[i]) == -1) { //Không tìm thấy
            b[*m] = a[i];
            (*m)++;
        }
    }

//Câu 8b
void TaoMangChuaCP(int a[], int n, int b[], int *m) {

```

```

    *m = 0;
    for(int i = 0; i < n; i++)
        if(NguyenTo(a[i])) {
            b[*m] = a[i];
            (*m)++;
        }
    }
}

```

❑ Bài tập luyện tập

9. Viết các hàm thực hiện các chức năng sau:

- Tính Tổng các phần tử có chữ số tận cùng là 6 và chia hết cho 6.
- Tính tổng các phần tử là số hoàn chỉnh/số nguyên tố/số đối xứng.

10. Viết các hàm thực hiện các chức năng sau:

- Đếm số phần tử có chữ số tận cùng là 6 và chia hết cho 6.
- Đếm số phần tử là số hoàn chỉnh/số nguyên tố/số đối xứng.
- Đếm số phần tử là số hoàn chỉnh/số nguyên tố/số đối xứng phân biệt, các phần tử là số hoàn chỉnh/số nguyên tố/số đối xứng trùng nhau chỉ đếm một lần.
- Đếm tần suất xuất hiện của các phần tử. Chẳng hạn với mảng gồm các phần tử: 12 34 12 34 43 12 5 thì tần suất xuất hiện các phần tử là 12 (3 lần), 34 (2 lần), 12 (2 lần), 5 (1 lần).

11. Viết các hàm thực hiện các chức năng sau:

- Kiểm tra mảng có gồm toàn số dương hay không?
- Kiểm tra mảng có thứ tự giảm hay không?
- Kiểm tra mảng có được sắp thứ tự hay không?
- Kiểm tra mảng có đan xen âm dương hay không?
- Kiểm tra mảng có tất cả bộ 3 phần tử đứng cạnh nhau lập thành cấp số cộng hay không?

12. Viết các hàm thực hiện các chức năng sau:

- Sắp xếp các phần tử là số hoàn chỉnh/số nguyên tố/số đối xứng tăng dần, các phần tử khác giữ nguyên vị trí.
- Trộn hai mảng tăng dần thành một mảng được sắp thứ tự tăng dần.

13. Viết hàm tìm vị trí của phần tử là số hoàn chỉnh/số nguyên tố/số đối xứng nhỏ nhất.

14. Viết các hàm thực hiện các chức năng sau:

- Xóa phần tử là số hoàn chỉnh/số nguyên tố/số đối xứng cuối cùng.
- Xóa tất cả phần tử là số hoàn chỉnh/số nguyên tố/số đối xứng.

15. Viết các hàm thực hiện các chức năng sau:

- a. Tạo mảng b từ mảng a sao cho mảng b chỉ chứa các phần tử là số hoàn chỉnh/số nguyên tố/ số đối xứng.
- b. Tạo mảng b từ mảng a sao cho mảng b chỉ chứa các phần tử là số hoàn chỉnh/số nguyên tố/ số đối xứng phân biệt.

CHƯƠNG 7: MẢNG HAI CHIỀU

Đối với các bài tập trong chương này, trong chương trình luôn phải có hai hàm nhập và xuất mảng hai chiều nguyên.

□ Bài tập có hướng dẫn

1. Viết các hàm thực hiện các chức năng sau:
 - a. Tính tổng các phần tử trên dòng thứ k.
 - b. Tính trung bình cộng các phần tử là số chính phương.

Hướng dẫn:

```
//Câu 1a
int Tong(int a[][SIZE2], int sc, int k) {
    int ret = 0;
    for(i = 0; i < sc; i++)
        ret += a[k][i];
    return ret;
}

//Câu 1b
float TBConng(int a[][SIZE2], int sd, int sc) {
    int dem = 0, s = 0;
    float ret;
    for(int i = 0; i < sd; i++)
        for(int j = 0; j < sc; j++)
            if(chinhPhuong(a[i][j])) {
                s += a[i][j];
                dem++;
            }
    if(dem > 0)
        ret = (float)s / dem;
    else ret = -1; //Ma trận không chứa số chính phương
    return ret;
}
```

2. Viết các hàm thực hiện các chức năng sau:
 - a. Đếm số phần tử là số chính phương.

- b. Đếm số phần tử là số chính phương trên dòng thứ k.

Hướng dẫn

```
//Câu 2a
int DemCP (int a[][SIZE2], int sd, int sc) {
    int ret = 0;
    int i, j;
    for(i = 0; i < sd; i++)
        for(j = 0; j < sc; j++) {
            if(chinhPhuong(a[i][j]))
                ret++;
        }
    return ret;
}

//Câu 2b
int DemCPTrenDong(int a[][SIZE2], int sc, int k) {
    int ret = 0;
    for(int i = 0; i < sc; i++) {
        if(chinhPhuong(a[k][i]))
            ret++;
    }
    return ret;
}
```

3. Viết các hàm thực hiện các chức năng sau:
- Kiểm tra ma trận có phần tử dương hay không.
 - Kiểm tra dòng thứ k của ma trận có tăng dần/giảm dần hay không.

Hướng dẫn:

```
//Câu 3a
int CoDuong(int a[][SIZE2], int sd, int sc) {
    int i, j;
    for(i = 0; i < sd - 1; i++)
        for(j = 0; j < sc - 1; j++) {
            if(a[i][j] > 0) {
                return 1;        //Ma trận có phần tử dương
            }
        }
}
```

```

    }

    return 0;    //Ma trận không có phần tử dương
}

//Câu 3b
int DongTang(int a[][SIZE2], int sc, int k) {
    int ret = 1; //Ma trận có dòng thứ k tăng
    for(int i = 0; i < sc - 1; i++)
        if(a[k][i] > a[k][i+1]) {
            ret = 0; //Ma trận có dòng thứ k không tăng
            break;
        }
    return ret;
}

```

4. Viết các hàm thực hiện các chức năng sau:

- Liệt kê các dòng có chứa ít nhất một phần tử là số chính phương.
- Liệt kê các dòng có chứa phần tử gồm toàn số chính phương.

Hướng dẫn:

```

//Câu 4a
void LietKeDongCoCP(int a[][SIZE2], int sd, int sc) {
    int cp; //Cờ
    int i, j;
    for(i = 0; i < sd; i++) {
        //Kiểm tra dòng thứ i có chứa số chính phương không?
        cp = 0;
        for(j = 0; j < sc; j++) {
            if(ChinhPhuong(a[i][j])) {
                cp = 1;
                break;
            }
        }
        //In dòng có số chính phương
        if(cp) {
            for(j = 0; j < sc; j++)
                printf("%d\t", a[i][j]);
            printf("\n");
        }
    }
}

```

```

    }
}
//Câu 4b
void LietKeDongToanCP(int a[][SIZE2], int sd, int sc) {
    int cp; //Cờ
    int i, j;
    for(i = 0; i < sd; i++) {
        /*Kiểm tra dòng thứ i có chứa các phần tử gồm toàn chính
        phương không? */
        cp = 1;
        for(j = 0; j < sc; j++) {
            if(ChinhPhuong(a[i][j]) == 0) {
                cp = 0;
                break;
            }
        }
        //In dòng có số chính phương
        if(cp) {
            for(j = 0; j < sc; j++)
                printf("%d\t", a[i][j]);
            printf("\n");
        }
    }
}

```

5. Viết các hàm thực hiện các chức năng sau:

- Tìm vị trí phần tử lớn nhất.
- Tìm vị trí phần tử lớn nhất dòng thứ k.
- Tìm vị trí phần tử dương nhỏ nhất dòng thứ k.

Hướng dẫn:

```

//Câu 5a
void TimVTMax(int a[][SIZE2], int sd, int sc, int *csdong, int *cscot) {
    int i, j;
    *csdong = 0; *cscot = 0;
    for(i = 0; i < sd; i++)

```

```

        for(j = 0; j < sc; j++)
            if(a[i][j] > a[*csdong][*cscot]) {
                *scdong = i;
                *cscot = j;
            }
    }
//Câu 5b
int TimVTMaxTrenDong(int a[][SIZE2], int sc, int k, int *csdong, int
                    *cscot) {
    int i, j;
    *scdong = k;
    *cscot = 0;
    for(i = 1; i < sc; i++) {
        if(a[k][i] > a[k][*cscot])
            *cscot = i;
    }
    return ret; //Phần tử max nằm ở dòng thứ ret (cột thứ k)
}

```

6. Viết các hàm thực hiện các chức năng sau:

- Sắp xếp các phần tử trên dòng thứ k tăng dần từ trái sang phải.
- Sắp xếp các cột của ma trận sao cho tổng giá trị của mỗi cột tăng dần từ trái sang phải.

Ma trận nhập vào

2	4	6	0
9	1	2	6

Σ **11 5 8 6**

Ma trận sau khi s.xếp

4	0	6	2
1	6	2	9

5 6 8 11

- Sắp xếp ma trận tăng dần trên mỗi dòng từ trái sang phải và từ trên xuống dưới.

Ma trận nhập vào

2	4	5	9
9	7	1	8
5	9	0	1
3	8	7	6

Ma trận sau khi s.xếp

0	1	1	2
3	4	5	5
6	7	7	8
8	9	9	9

Hướng dẫn:

```

//Câu 6a
void SapXepDong(int a[][SIZE2], int sc, int k) {

```

```

int i, j;
for(i = 0; i < sc-1; i++)
    for(j = i+1; j < sc; j++)
        if(a[k][i] > a[k][j]) {
            int tam = a[k][i];
            a[k][i] = a[k][j];
            a[k][j] = tam;
        }
}

//Câu 6b
void HoanViCot(int a[][SIZE2], int sd, int i, int j) {
    for(int k = 0; k < sd; k++) {
        int tam = a[k][i];
        a[k][i] = a[k][j];
        a[k][j] = tam;
    }
}

void SXTongCotTang(int a[][SIZE2], int sd, int sc) {
    for(int i = 0; i < sc-1; i++)
        for(int j = i+1; j < sc; j++) {
            int s1 = TongCot(a, sd, i);    //Tổng các phần tử trên cột i
            int s2 = TongCot(a, sd, j);    //Tổng các phần tử trên cột j
            if(s1 > s2)
                HoanViCot(a, sd, i, j);    //Hoán vị cột i và cột j
        }
}

//Câu 6c
void SapXepCau6d(int a[][SIZE2], int sd, int sc) {
    int b[SIZE1*SIZE2];
    int m = 0;
    // Xây dựng mảng một chiều b từ mảng hai chiều a cho trước
    for(i = 0; i < sd; i++)
        for(j = 0; j < sc; j++) {
            b[m] = a[i][j]; m++;
        }
    // Sắp xếp mảng một chiều b tăng dần

```

```
SapXepTang(b, m);
// Đặt các phần tử trong b ngược trở về a
m = 0;
for(i = 0; i < sd; i++)
    for(j = 0; j < sc; j++) {
        a[i][j] = b[m]; m++;
    }
}
```

□ Bài tập luyện tập

7. Viết các hàm thực hiện các chức năng sau:
 - a. Tính tổng/tích các phần tử trên cột thứ k.
 - b. Tính trung bình cộng các phần tử là số hoàn chỉnh/số nguyên tố/số đối xứng.
8. Viết các hàm thực hiện các chức năng sau:
 - a. Đếm số phần tử là số hoàn chỉnh/số nguyên tố/số đối xứng.
 - b. Đếm số phần tử là số hoàn chỉnh/số nguyên tố/số đối xứng trên cột thứ k.
9. Viết các hàm thực hiện các chức năng sau:
 - a. Kiểm tra ma trận có gồm toàn phần tử dương hay không.
 - b. Kiểm tra cột thứ k của ma trận có tăng dần/giảm dần hay không.
10. Viết các hàm thực hiện các chức năng sau:
 - a. Liệt kê các cột có chứa phần tử là số hoàn chỉnh/số nguyên tố/số đối xứng.
 - b. Liệt kê các cột có chứa phần tử gồm toàn là số chính phương/số hoàn chỉnh/số nguyên tố/số đối xứng.
11. Viết các hàm thực hiện các chức năng sau:
 - a. Tìm vị trí phần tử nhỏ nhất.
 - b. Tìm vị trí phần tử nhỏ nhất trên cột thứ k.
 - c. Tìm vị trí phần tử âm lớn nhất trên cột thứ k.
12. Viết các hàm thực hiện các chức năng sau:
 - a. Sắp xếp các phần tử trên cột thứ k tăng dần từ trên xuống dưới.
 - b. Sắp xếp các dòng của ma trận sao cho tổng giá trị của mỗi dòng tăng dần từ trên xuống dưới.
13. Viết các hàm thực hiện các chức năng sau trên mảng vuông
 - a. Đếm các phần tử là số chính phương/số hoàn chỉnh/số nguyên tố/số đối xứng nằm trên đường chéo phụ.

- b. Đếm các phần tử là số chính phương/số hoàn chỉnh/số nguyên tố/số đối xứng nằm trong nửa mảng vuông phía dưới đường chéo chính/nửa mảng vuông phía trên đường chéo phụ/ nửa mảng vuông phía dưới đường chéo phụ.
- c. Tìm phần tử lớn nhất/nhỏ nhất trên đường chéo phụ.
- d. Tìm phần tử âm lớn nhất/dương nhỏ nhất trên đường chéo phụ.
- e. Sắp xếp các phần tử trên đường chéo phụ theo thứ tự tăng/giảm dần từ trên hướng xuống/từ dưới hướng lên.

CHƯƠNG 8: CHUỖI

❑ Bài tập có hướng dẫn

- Viết chương trình kiểm tra một chuỗi có đối xứng hay không?

Ví dụ: Các chuỗi level, radar, dad, ... là các chuỗi đối xứng

Hướng dẫn

```
int DoiXung(char *s) {
    int len = strlen(s);
    int ret = 1;
    for(int i = 0; i <= len/2; i++)
        if(s[i] != s[len - 1 - i]) {
            ret = 0; break;
        }
    return ret;
}
```

- Viết chương trình tìm chuỗi đảo ngược của một chuỗi.

Ví dụ: Chuỗi "Lap trinh C" có chuỗi đảo ngược là "C hni rt paL"

Hướng dẫn

```
//Dữ liệu đầu vào là s1, dữ liệu đầu ra là s2 đảo ngược
void TimDaoNguoc(char *s1, char *s2) {
    int len = strlen(s1);
    int j = 0;
    for(int i = len - 1; i >= 0; i--) {
        s2[j] = s1[i];
        j++;
    }
    s2[j] = 0; //Thêm ký tự '\0' và cuối chuỗi s2
}
```

- Viết chương trình đếm số từ trong một chuỗi. Qui ước các từ trong chuỗi cách nhau ít nhất một khoảng trắng.

Ví dụ: Chuỗi " Lap trinh C " có 3 từ

Hướng dẫn

```
int DemTu(char *s) {
```



```

int len = strlen(s);
int ret = 0;
int i = 0;
while(i < len) {
    //Đi qua các ký tự trắng
    while(s[i] == ' ' && i < len)
        i++;
    if(i < len) {
        //Đi qua các ký tự khác trắng
        while(s[i] != ' ' && i < len)
            i++;
        ret++;
    }
}
return ret;
}

```

4. Viết chương trình in mỗi từ của chuỗi trên một dòng.

Ví dụ: Chuỗi " Lập trình C" sẽ được in ra:

Lập
trình
C

```

void InTu(char *s) {
    int i = 0, j;
    char tu[20];
    int len = strlen(s);
    while(i < len) {
        //Đi qua các khoảng trắng
        while(s[i] == ' ' && i < len)
            i++;
        if(i < len) {
            //Đi qua các ký tự khác trắng
            j = 0;
            while(s[i] != ' ' && i < len) {
                tu[j] = s[i];
                j++; i++;
            }
        }
    }
}

```

```

    }
    tu[j] = 0;
    printf("%s\n" tu);
}
}
}

```

5. Viết chương trình đếm số lần xuất hiện của mỗi ký tự từ A đến Z có trong một chuỗi (không phân biệt chữ hoa hay chữ thường).

Ví dụ: Với chuỗi "Hello World" ta sẽ có:

D: 1 lần

E: 1 lần

H: 1 lần

L: 3 lần

...

```

#include <ctype.h>
#define NUM_ALPHABET 26 //Số ký tự trong bảng chữ cái
void ThongKe(char *s) {
    int dem[NUM_ALPHABET]; //Mảng đếm số lần xuất hiện ký tự
    //Khởi tạo mảng đếm
    for(i = 0; i < NUM_ALPHABET; i++)
        dem[i] = 0;
    //Đếm số lần xuất hiện ký tự
    int len = strlen(s);
    for(i = 0; i < len; i++)
        if(toupper(s[i]) >= 'A' && toupper(s[i]) <= 'Z')
            dem[toupper(s[i]) - 'A']++;
    //In kết quả
    for(i = 0; i < NUM_ALPHABET; i++)
        printf("ký tự %c: %d lan\n", i + 'A', dem[i]);
}

```

6. Viết một hàm stringcpy có chức năng tương tự như hàm **strcpy** của thư viện.

Hướng dẫn

```

void stringcpy(char *s1, char *s2) {
    int i = 0;

```

```
while((s1[i] = s2[i]) != 0)
    i++;
}
```

□ Bài Tập luyện tập

7. Viết chương trình xoá đi ký tự 'e' có trong một chuỗi nhập từ bàn phím.

Ví dụ: chuỗi ban đầu = "Que hương là chum khe ngọt"
 chuỗi sau khi xoá = "Qu hương là chum kh ngọt"

8. Viết chương trình xoá từ đầu tiên có trong một chuỗi nhập từ bàn phím.

Ví dụ: chuỗi ban đầu = "Que hương là chum khe ngọt"
 chuỗi sau khi xoá = "hương là chum khe ngọt"

9. Viết một hàm stringlen có chức năng tương tự như hàm **strlen** của thư viện.
 10. Viết một hàm stringcmp có chức năng tương tự như hàm **strcmp** của thư viện.
 11. Viết một hàm stringcat có chức năng tương tự như hàm **strcat** của thư viện.
 12. Viết một hàm stringchar có chức năng tương tự như hàm **strchar** của thư viện.
 13. Viết một hàm stringstring có chức năng tương tự như hàm **strchar** của thư viện.

CHƯƠNG 9: CẤU TRÚC

□ Bài tập có hướng dẫn

1. Khai báo kiểu dữ liệu biểu diễn thông tin của một phân số, sau đó viết chương trình thực hiện các chức năng sau:
 - a. Nhập phân số.
 - b. Xuất phân số.
 - c. Rút gọn phân số.
 - d. Tính tổng hai phân số.
 - e. Tính hiệu hai phân số.
 - f. Tính tích hai phân số.
 - g. Tính Thương hai phân số.
 - h. So sánh hai phân số.
 - i. Kiểm tra phân số âm.
 - j. Kiểm tra phân số dương.

Hướng dẫn:

```
struct PS {
    int ts, ms;
};
//Câu 1c
void RutGon (PS *u) {
    int uscln = TimUSCLN(u ->ts, u->ms);
    u->ts /= uscln;
    u->ms /= uscln;
}
//Câu 1d
PS Tong(PS u, PS v) {
    PS ret;
    ret.ts = u.ts * v.ms + u.ms * v.ts;
    ret.ms = u.ms * v.ms;
    RutGon(&ret);
    return ret;
}
//Câu 1h
```

```
int SoSanh(PS u, PS v) {
    int ret;
    if(u.ts * v.ms > u.ms * v.ts)
        ret = 1; //u lớn hơn v
    else if(u.ts * v.ms == u.ms * v.ts)
        ret = 0; //u bằng v
    else ret = -1; //u nhỏ hơn v
    return ret;
}
```

2. Khai báo kiểu dữ liệu biểu diễn thông tin điểm trong mặt phẳng, sau đó viết chương trình thực hiện các chức năng sau:
- Nhập điểm
 - Xuất điểm theo định dạng (x, y)
 - Tính khoảng cách giữa hai điểm
 - Tính khoảng cách giữa hai điểm theo phương Ox
 - Tính khoảng cách giữa hai điểm theo phương Oy
 - Tìm điểm đối xứng qua gốc tọa độ
 - Tìm điểm đối xứng qua trục hoành
 - Tìm điểm đối xứng qua trục tung
 - Tìm điểm đối xứng qua đường phân giác thứ nhất ($y=x$)
 - Tìm điểm đối xứng qua đường phân giác thứ hai ($y=-x$)
 - Kiểm tra điểm có thuộc phần tư thứ I không?
 - Kiểm tra điểm có thuộc phần tư thứ II không?
 - Kiểm tra điểm có thuộc phần tư thứ III không?
 - Kiểm tra điểm có thuộc phần tư thứ IV không?

Hướng dẫn:

```
struct Diem {
    int x, y;
};
//câu 2a
void Nhap1Diem(Diem *u) {
    printf("Nhap hoành do:"); scanf("%d", &u->x);
    printf("Nhap tung do:"); scanf("%d", &u->y);
}
//Câu 2b
```

```

void Xuat1Diem(Diem u) {
    printf("(%d, %d)\n", u.x, u.y);
}

//Câu 2c
float KhoangCach(Diem u, Diem v) {
    return sqrt(pow(u.x - v.x, 2) + pow(u.y - v.y, 2));
}

//Câu 2f
Diem DoiXungQuaO(Diem u) {
    Diem ret;
    ret.x = -u.x;
    ret.y = -u.y;
    return ret;
}

//câu 2i
Diem DoiXungQuaPhanGiac1(Diem u) {
    Diem ret;
    ret.x = u.y;
    ret.y = u.x;
    return ret;
}

//Câu 2k
int DiemThuocPhanTu1(Diem u) {
    if(u.x > 0 && u.y > 0)
        return 1; //Thuộc
    else return 0; //Không thuộc
}
    
```

3. Khai báo kiểu dữ liệu để biểu diễn thông tin của một chuyến bay, biết rằng một chuyến bay gồm các thành phần: Mã chuyến bay, ngày bay, giờ bay, nơi đi, nơi đến. Sau đó viết chương trình nhập và xuất cho kiểu dữ liệu này.

Hướng dẫn :

```

struct NGAY{
    int d, m, y;
};

struct GIO {
    
```

```

    int h, m, s;
};
struct CB {
    char macb[10];
    NGAY ngaybay;
    GIO giobay;
    char noidi[20];
    char noden[20];
};
void NhapNgay(NGAY *u) {
    printf("\tNhap ngay:"); scanf("%d", &u->d);
    printf("\tNhap thang:"); scanf("%d", &u->m);
    printf("\tNhap nam:"); scanf("%d", &u->y);
}
void XuatNgay(NGAY u) {
    printf("%02d - %02d - %02d\n", u.d, u.m, u.y);
}
void NhapGio(GIO *u) {
    printf("\tNhap gio:"); scanf("%d", &u->h);
    printf("\tNhap phut:"); scanf("%d", &u->m);
    printf("\tNhap giay:"); scanf("%d", &u->s);
}
void XuatNgay(GIO u) {
    printf("%02d : %02d : %02d\n", u.h, u.m, u.s);
}
void Nhap1CB(CB *u) {
    printf("Nhap ma chuyen bay:");
    gets(u->macb) ;
    printf("Nhap ngay bay: ") ;
    NhapNgay(&u->ngaybay) ;
    printf("Nhap gio bay: ") ;
    NhapGio(&u->giobay) ;
    printf("Nhap noi di:");
    gets(u->noidi);
    printf("Nhap noi den:");

```

```

        gets(u->noiden);
    }
    void Xuat1CB(CB u) {
        printf("%s\t%02d-%02d-%02d\t%02d:%02d\t%s\t%s\n",
               u.macb, u.ngaybay.d, u.ngaybay.m, u.ngaybay.y, u.giobay.h,
               u.giobay.s, u.noidi, u.noiden) ;
    }

```

4. Viết chương trình thực hiện các chức năng sau:

- a. Nhập danh sách điểm
- b. Xuất danh sách điểm
- c. Đếm số lượng điểm có hoành độ dương
- d. Tìm một điểm có tung độ lớn nhất trong mảng
- e. Tìm một điểm trong mảng gần gốc tọa độ nhất
- f. Tìm hai điểm gần nhau nhất trong mảng
- g. Tìm hai điểm xa nhau nhất trong mảng

Hướng dẫn:

```

//Câu 11e
int TimDiemGanONhat(DSD u) {
    Diem o = {0, 0}; //Gốc tọa độ
    int ret = 0;
    float kc1 = KhoangCach(u.arr[ret], o);
    for(int i = 1; i < u.n; i++) {
        float kc2 = KhoangCach(u.arr[i], o);
        if(kc2 < kc1) {
            ret = i;
            kc1 = kc2;
        }
    }
    return ret; //Trả về vị trí điểm gần gốc o nhất
}

//Câu 11 f (Hàm trả về vị trí hai điểm gần nhau nhất)
void Tim2DiemGanNhanhNhat(DSD u, int *vt1, int *vt2) {
    *vt1 = 0; *vt2 = 1;
    float kc1 = KhoangCach(u.arr[*vt1], u.arr[*vt2]);
    float kc2;

```



```

for(int i = 0; i < u.n - 1; i++)
    for(int j = i + 1; j < u.n; j++) {
        kc2 = KhoangCach(u.arr[i], u.arr[j]);
        if(kc2 < kc1) {
            *vt1 = i;
            *vt2 = j;
            kc1 = kc2;
        }
    }
}

```

5. Viết chương trình thực hiện các chức năng sau:
- Nhập danh sách phân số.
 - Xuất danh sách phân số.
 - Rút gọn danh sách phân số.
 - Tìm phân số dương nhỏ nhất/phân số âm lớn nhất.
 - Sắp xếp danh sách phân số theo thứ tự mẫu số giảm dần.

Hướng dẫn

```

//Câu 1c
void RutGonDSPS(DSPS *u) {
    int i;
    for(i = 0; i < u->n; i++)
        RutGon(&u->arr[i]);
}

//Câu 1d
int TimPSDuongMin(DSPS u) {
    int i, j, ret;
    //Tìm vị trí phân số dương đầu tiên
    for(i = 0; i < u->n; i++)
        if(u.arr[i].ts * u.arr[i].ms > 0) break;
    if(i >= u->n) //Không có phân số dương
        ret = -1;
    else { //Có phân số dương
        //Tìm vị trí phân số dương min
        ret = i;
        for(j = i + 1; j < u->n; j++)

```

```

                if(u.arr[j].ts * u.arr[j].ms > 0 && SoSanhPS(u.arr[j],u.arr[ret]
                == -1))
                    ret = j;
            }
            return ret;
        }
    
```

6. Viết chương trình thực hiện các chức năng sau:

- a. Nhập danh sách chuyến bay.
- b. Xuất danh sách chuyến bay.
- c. Sắp xếp danh sách chuyến bay tăng dần theo ngày bay.

Hướng dẫn:

```

struct DSCB {
    int n;                //Số chuyến bay
    CB arr[SIZE];        //Danh sách chuyến bay
};

//Câu 6a
voidNhapDSCB(DSCB *u) {
    int i;
    do {
        printf("Nhap so chuyen bay:");
        scanf("%d", &u->n);
    } while(u->n < 1 || u->n > SIZE);
    for(i = 0; i < u->n; i++) {
        printf("Chuyen bay thu %d", i);
        Nhap1CB(&u->arr[i]);
    }
}

//Câu 6b
void XuatDSCB(DSCB u) {
    for(i = 0; i < u.n; i++)
        Xuat1CB(u.arr[i]);
}

//Câu 6c
int SoSanhNgay(NGAY u, NGAY v) {
    //Trả về 1: u > v, 0: u = v, -1: u < v
    
```

```

int ret;
if(u.y > v.y)
    ret = 1;
else if(u.y < v.y)
    ret = -1;
else {
    if(u.m > v.m)
        ret = 1;
    else if(u.m < v.m)
        ret = -1;
    else {
        if(u.d > v.d)
            ret = 1;
        else if(u.d < v.d)
            ret = -1;
        else ret = 0;
    }
}
return ret;
}

void SapXepDSCB(DSCB *u) {
    for(i = 0; i < u->n - 1; i++)
        for(j = i + 1; j < u->n; j++)
            if(SoSanhNgay(u->arr[i].ngaybay, u->arr[j].ngaybay) == 1) {
                CB tam = arr[i];
                arr[i] = arr[j];
                arr[j] = tam;
            }
}

```

❑ Bài tập luyện tập

7. Khai báo kiểu dữ liệu biểu diễn thông tin của một số phức, sau đó viết chương trình thực hiện các chức năng sau:
 - a. Nhập số phức
 - b. Xuất số phức (theo định dạng $Re + Im*i$)

- c. Tính modul số phức
 - d. So sánh hai số
 - e. Tính tổng hai số phức
 - f. Tính hiệu hai số phức
 - g. Tính tích hai số phức
 - h. Tính thương hai số phức
8. Khai báo kiểu dữ liệu biểu diễn thông tin đường tròn, sau đó viết chương trình thực hiện các chức năng sau:
- a. Nhập đường tròn
 - b. Xuất đường tròn theo định dạng $((x, y), r)$
 - c. Tính chu vi đường tròn
 - d. Tính diện tích đường tròn
 - e. Xét vị trí tương đối giữa hai đường tròn (trùng nhau, ở trong nhau, tiếp xúc, cắt nhau, nằm ngoài nhau)
 - f. Kiểm tra một điểm có nằm trên đường tròn không?

Nhắc lại:

Nếu $kc(o1, o2) = 0$ và $o1 = o2$ thì hai đường tròn trùng nhau.

Nếu $abs(r1 - r2) > kc(o1, o2)$ thì hai đường tròn ở trong nhau.

Nếu $r1 + r2 = kc(o1, o2)$ hoặc $abs(r1 - r2) = kc(o1, o2)$ thì hai đường tròn tiếp xúc nhau.

Nếu $r1 + r2 > kc(o1, o2)$ và $abs(r1 - r2) < kc(o1, o2)$ thì hai đường tròn cắt nhau.

Nếu $abs(r1 - r2) < kc(o1, o2)$ thì hai đường tròn ở ngoài nhau

9. Khai báo kiểu dữ liệu biểu diễn thông tin một tam giác, sau đó viết chương trình thực hiện các chức năng sau:
- a. Nhập tam giác
 - b. Xuất tam giác theo định dạng $((x1, y1); (x2, y2); (x3, y3))$
 - c. Kiểm tra 3 đỉnh có lập thành 3 đỉnh của một tam giác không?
 - d. Tính chu vi tam giác
 - e. Tính diện tích tam giác
 - f. Tính tọa độ trọng tâm tam giác
 - g. Phân loại tam giác
 - h. Tìm một đỉnh có hoành độ lớn nhất
 - i. Tìm một đỉnh có tung độ nhỏ nhất
 - j. Tính tổng khoảng cách từ điểm $p(x, y)$ tới 3 đỉnh tam giác.
10. Khai báo kiểu dữ liệu biểu diễn thông tin ngày, sau đó viết chương trình thực hiện các chức năng sau:
- a. Nhập ngày

- b. Xuất ngày theo định dạng (dd/mm/yy)
 - c. Kiểm tra ngày hợp lệ
 - d. Tìm ngày kế tiếp
 - e. Tìm ngày trước đó.
11. Khai báo kiểu dữ liệu biểu diễn thông tin giờ, sau đó viết chương trình thực hiện các chức năng sau:
 - a. Nhập giờ
 - b. Xuất giờ theo định dạng (hh:mm:ss)
 - c. Kiểm tra giờ hợp lệ
 - d. Tính tổng hai giờ
12. Khai báo kiểu dữ liệu để biểu diễn thông tin của một sinh viên, biết rằng một sinh viên bao gồm các thành phần: Mã sinh viên, Tên sinh viên, Ngày sinh, Điểm. Sau đó viết chương trình nhập xuất cho kiểu dữ liệu này.
13. Khai báo kiểu dữ liệu để biểu diễn thông tin của một phiếu hàng, biết rằng một phiếu hàng bao gồm các thành phần: Người mua, Mã hàng, Đơn giá, Số lượng. Sau đó viết chương nhập xuất cho kiểu dữ liệu này.
14. Viết chương trình thực hiện các chức năng sau:
 - a. Nhập danh sách sinh viên
 - b. Xuất danh sách sinh viên
 - c. Tìm và in các sinh viên có điểm cao nhất
 - d. Sắp xếp danh sách sinh viên tăng dần theo mã.
15. Viết chương trình thực hiện các chức năng sau:
 - a. Nhập danh sách phiếu hàng
 - b. Xuất danh sách phiếu hàng
 - c. Tính tổng số tiền bán được theo từng mã hàng.

CHƯƠNG 10: CON TRỎ

Làm lại các bài tập về mảng một chiều (chương 6), mảng hai chiều (chương 7) và chuỗi (chương 8) bằng cách sử dụng con trỏ.

Hướng dẫn bài 1 chương 6:

```
//Câu 1a
int ChuSoDauTienLe(int x) {
    while (x >= 10)
        x /= 10 ;
    return (x % 2 != 0 ? 1 : 0) ;
}

int TongChuSoDauTienLe(int a[], int n) {
    int ret = 0;
    for(i = 0; i < n; i++)
        if(ChuSoDauTienLe(*(a + i)))
            ret += a[i];

    return ret;
}

//Câu 1b
int TongChinhPhuong(int a[], int n) {
    int ret = 0;
    for(i = 0; i < n; i++)
        if(ChinhPhuong(*(a + i)))
            ret += *(a + i);

    return ret;
}
```

Hướng dẫn bài 1 chương 7:

```
//Câu 1a
int Tong(int a[][SIZE2], int sc, int k) {
    int ret = 0;
    for(i = 0; i < sc; i++)
        ret += (*(a + k) + i);

    return ret;
}

//Câu 1b
```

```
float TBCong(int a[][SIZE2], int sd, int sc) {
    int dem = 0, s = 0;
    float ret;
    for(int i = 0; i < sd; i++)
        for(int j = 0; j < sc; j++)
            if(chinhPhuong(*(a + i) + j)) {
                s += *(a + i) + j;
                dem++;
            }
    if(dem > 0)
        ret = (float)s / dem;
    else ret = -1; //Ma trận không chứa số chính phương
    return ret;
}
```

Hướng dẫn bài 1 chương 8:

```
int DoiXung(char *s) {
    int len = strlen(s);
    int ret = 1;
    for(int i = 0; i <= len/2; i++)
        if(*(s + i) != *(s + (len - 1 - i))) {
            ret = 0; break;
        }
    return ret;
}
```