

```
1  #include <stdio.h>
2  #include <math.h>
3  void nhap(float a[], int &n) {
4      for (int i=0; i< n; i++){
5          printf ("a[%d] = ",i);
6          scanf ("%f", a + i);
7      }
8  }
9  double sum(float a[], int &n){
10     double sum =0;
11     for (int i=0; i< n; i++){
12         sum += *(a+i);
13     }return sum;
14 }
15 int main () {
16     float a[100];
17     int n;
18     scanf ("%d",&n);
19     nhap(a, n);
20     printf (" tong la %2.f",sum(a, n));
21 }
```

```
1  #include <stdio.h>
2  #include <math.h>
3  void nhap(float *a, int n) {
4      for (int i=0; i< n; i++){
5          printf ("a[%d] = ",i);
6          scanf ("%f", &a[i]);
7      }
8  }
9  int doixung(float *a, int n){
10     int k;
11     for (int i=0; i< n/2; i++){
12         if ( a[i] != a[n-1-i])
13             k =1;
14         else k = 0;
15     } return k;
16 }
17 int main () {
18     float a[100];
19     int n;
20     scanf ("%d",&n);
21     nhap(a, n);
22     doixung(a, n);
23     if (doixung(a, n) == 1)
24         printf (" mang nay khong doi xung ");
25     else printf (" mang nay doi xung");
26 }
```

```

1  #include <stdio.h>
2  #include <math.h>
3  // thuat toan sap xep
4  void nhap(float a[], int n) {
5      for (int i=0; i< n; i++){
6          printf ("a[%d] = ",i);
7          scanf ("%f", &a[i]);
8      }
9  }
10 void swap(float &a, float &b){
11     int temp =a;
12     a=b;
13     b=temp;
14 }
15 void bubblesort(float a[], int n){
16     for (int i = 0; i < n - 1; i++)
17         for (int j = 0; j < n - i - 1; j++)
18             if (a[j] > a[j + 1])
19                 swap(a[j], a[j + 1]);
20 }
21 void selectsort(float a[], int n)
22 {
23     int min;
24     for (int i = 0; i < n - 1; i++)
25     {
26         min = i; // coi a[i] min
27         for (int j = i + 1; j < n; j++) // tim min tu a[i] den a[n-1]
28             if (a[j] < a[min]) // a[j] nho hon a[min] thi a[j] la nho nhat
29                 min = j; // luu vi tri min moi tim dc
30         if (min != i) // neu min moi nay ko gion ban dau thi swap
31             swap(a[i], a[min]);
32     }
33 }

```

```
21 void selectsort(float a[], int n)
22 {
23     int min;
24     for (int i = 0; i < n - 1; i++)
25     {
26         min = i; // coi a[i] min
27         for (int j = i + 1; j < n; j++) // tim min tu a[i] den a[n-1]
28             if (a[j] < a[min]) // a[j] nho hon a[min] thi a[j] la nho nhat
29                 min = j; // luu vi tri min moi tim dc
30         if (min != i) // neu min moi nay ko gion ban dau thi swap
31             swap(a[i], a[min]);
32     }
33 }
34 void insertsort(float a[], int n){
35     int pos, x;
36     for (int i = 1; i < n; i++){
37         x = a[i]; // luu lai x de tranh bi ghi de khi dich chuyen phan tu
38         pos = i - 1;
39         // tim vitri de chen x
40         while (pos >= 0 && a[pos] > x){
41             // ket hop dich chuyen phan tu de nhuong cho cho x
42             a[pos + 1] = a[pos];
43             pos--;
44         }
45         // chen x vao vi tri tim dc
46         a[pos + 1] = x;
47     }
48 }
```

```

49 void binsort(float a[],int n ){
50     int l,r,m,i;
51     int x;//luu lai x de tranh bi ghi de khi dich chuyen phan tu
52     for(int i=1 ; i<n ; i++){
53         x = a[i]; l = 0;
54         r = i-1;
55         while(l<=r) { // tim vi tri chen x {
56             m = (l+r)/2;
57             // tim vi tri thich hop m
58             if(x < a[m]) r = m-1;
59             else l = m+1;
60         }
61         for(int j = i-1 ; j >=l ; j--)
62             a[j+1] = a[j]; // doi cac phan tu ra sau x
63         a[l] = x; // chen x
64     }
65 }
66 void xuat(float a[], int n){
67     for(int i=0;i<n;i++){
68         printf (" %2.f ", *(a+i));
69     }


```

```
--  
70 int main ()  
71 {  
72     float a[100];  
73     int n;  
74     scanf ("%d",&n);  
75     nhap(a, n);  
76     bubblesort(a, n);  
77     printf (" sap xep kieu bubblesort ");  
78     xuat( a, n);  
79     selectsort(a, n);  
80     printf ("\n sap xep kieu selectsort ");  
81     xuat( a, n);  
82     insertsort(a, n);  
83     printf ("\n sap xep kieu insertsort ");  
84     xuat( a, n);  
85     bininsertionsort(a,n );  
86     printf ("\n sap xep kieu bininsertionsort ");  
87     xuat( a, n);  
88 }
```

1. Gán $i = 0$
2. Gán $j = 0$
3. Nếu $A[j] > A[j + 1]$ thì đổi chỗ $A[j]$ và $A[j + 1]$
4. Nếu $j < n - i - 1$:
 - Đúng thì $j = j + 1$ và quay lại bước 3
 - Sai thì sang bước 5
5. Nếu $i < n - 1$:
 - Đúng thì $i = i + 1$ và quay lại bước 2
 - Sai thì dừng lại

1. Gán $i = 0$
2. Gán $j = i + 1$ và $\text{min} = A[i]$
3. Nếu $j < n$:
 - Nếu $A[j] < A[\text{min}]$ thì $\text{min} = j$
 - $j = j + 1$
 - Quay lại bước 3
4. Đổi chỗ $A[\text{min}]$ và $A[i]$
5. Nếu $i < n - 1$:
 - Đúng thì $i = i + 1$ và quay lại bước 2
 - Sai thì dừng lại



- 
1. Gán $i = 1$
 2. Gán $x = A[i]$ và $pos = i - 1$
 3. Nếu $pos \geq 0$ và $A[pos] > x$:
 - $A[pos + 1] = A[pos]$
 - $pos = pos - 1$
 - Quay lại bước 3
 4. $A[pos + 1] = x$
 5. Nếu $i < n$:
 - Đúng thì $i = i + 1$ và quay lại bước 2
 - Sai thì dừng lại

```

1  #include<stdio.h>
2  #include<math.h>
3  #include<string.h>
4  // bai 2 tr 46 ltnc
5  void nhap(int a[], int n) {
6      for (int i=0; i< n; i++){
7          printf ("a[%d] = ",i);
8          scanf ("%f", a + i);
9      }
10 }
11 double trungbinhcong(int a[], int n){
12     int t=0,dem=0;
13     for(int i=0 ; i<n ; i++){
14         t+=*a++;
15         dem+=1;
16     } return t/dem ;
17 }
18 int max(int a[], int n){
19     int max=a[0];
20     for(int i=0 ; i<n ; i++){
21         if ( max<a[i] )
22             max=a[i];
23     } return max;
24 }
25 void nhohonx(int a[], int n, int x){
26     for (int i=0 ; i<n ; i++){
27         if ( a[i] < x && a[i] % 2==0)
28             printf ("%f", a[i]);
29     }
30 }
31 int daydondieu(int a[], int n){
32     int dem1=0;
33     for(int i=0; i<n ; i++){
34         if(a[i] <=a[i+1])
35             dem1+=1;
36     } return dem1;

```

```

37 }
38 void xuat(int a[], int n){
39     printf("\nday so ban vua nhap la:\n");
40     for(int i=0 ; i<n ; i++){
41         printf("%2d",a[i]);
42     }
43 }
44 void daonguocday(int a[], int n){
45     printf("\nday so moi la:\n");
46     for(int i=1; i<=n ; i++){
47         printf("%2f",a[n-i]);
48     }
49 }
50 void chantrlesau(int a[], int n, int b[]){
51     int dem2=0;
52     for(int i=0 ; i<n ; i++){
53         if(a[i]%2==0){
54             b[dem2]=a[i];
55             dem2+=1;
56         }
57     }
58     for(int i=0 ; i<n ; i++){
59         if(a[i]%2 != 0){
60             b[dem2]=a[i];
61             dem2+=1;
62         }
63     }
64     printf("\nday moi cua ban la:\n ");
65     for(int i=0 ; i<n ; i++){
66         printf("%d",b[i]);
67     }
68 }

```

```
69 int main (){
70     int a[50],b[50];
71     int x,n; int dem1, dem2;
72     printf("nhap so phan tu cua day: ");
73     scanf("%d",&n);
74     printf("nhap x:");
75     scanf("%d",&x);
76     nhap(a, n);
77     printf("trung binh cong cua day so la: %f",trungbinhcong(a, n));
78     printf( "\n gia tri lon nhat cau day so la: %d ", max(a, n));
79     printf ( " \nphan tu chan nho hon x : ");
80     nhohonx(a, n,x);
81     if(daydondieu(a, n)==n-1) {
82         printf("\nday la day don dieu khong giam");
83     }
84     else {
85         printf("\nday khong phai la day don dieu khong giam");
86     }
87     xuat(a, n);
88     daonguocday(a, n);
89     chantrlesau(a, n, b);
90
91 }
92
```

```
1  #include<stdio.h>
2  #include<math.h>
3  #include<string.h>
4  //bai 3 tr 46
5  void nhap( float x[], float y[], int n){
6      for(int i=0 ; i<n ; i++){
7          printf("nhap diem toa do x[%d]y[%d]:",i,i);
8          scanf("%f%f",x+i,y+i);
9      }
10 }
11 double gapkhuc( float x[], float y[], int n){
12     double t = 0;
13     for(int i=0 ; i<n ; i++){
14         t+=sqrt((pow((x[i]-x[i+1]),2))+pow((y[i]-y[i+1]),2));
15     }
16     return t;
17 }
18 void xatruchoanh( float *x, float *y, int n){
19     float max=0;
20     for(int i=0 ; i<n ; i++){
21         if(max<abs(y[i]))
22             max= abs(y[i]);
23     }
```

```

18 void xatruchoanh( float *x, float *y, int n){
19     float max=0;
20     for(int i=0 ; i<n ; i++){
21         if(max<abs(y[i]))
22             max= abs(y[i]);
23     }
24     printf("\n diem xa truc hoan nhac la:");
25     for(int i=0 ; i<n ; i++){
26         if(abs(y[i])==max)
27             printf("\n(%f,%f)",x[i],y[i]);
28     }
29 }
30 int diemtrongdtr (float x[], float y[], int n){
31     int dem=0;
32     for(int i=0 ; i<n ; i++){
33         if( sqrt(( pow((x[i]-x[1]),2))+pow((y[i]-y[1]),2)) < 10)
34             dem+=1;
35     } return dem;
36 }
37 int main (){
38     //nhap du lieu
39     float x[50],y[50];int n;
40     scanf("%d",&n);
41     printf("nhap so diem tren mat phang xOy : ");
42     nhap( x,y, n);
43     // gap khuc
44     printf("\ndo dai duong gap khuc la: %2.f",gapkhuc(x, y,n));
45     // diem xa truc hoan
46     xatruchoanh(x,y,n);
47     // tam A1 ban kinh 10
48     printf("\nsố diem nam trong duong tron tam A1 ban kinh 10 la: %d",diemtrongdtr(x,y, n));
49
50 }
51

```

```

1  #include<stdio.h>
2  #include<string.h>
3  // bai 5 chuoai ky tu
4  void nhap(char *s){
5      printf("nhap chuoai: ");
6      gets(s);
7  }
8  void hoasaucham(char *s){
9      for (int i = 0; i < strlen(s); i++ ){
10         if (s[i] == '.')
11             if (s[i+1] >='a' && s[i+1] <='z')
12                 s[i+1] -=32;
13     }
14 }
15 void thuongsauphay(char *s){
16     for (int i = 0; i < strlen(s); i++ ){
17         if (s[i] == ',')
18             if (s[i+1] >='A' && s[i+1] <='Z')
19                 s[i+1] +=32;
20     }
21 }
22 void xoa(char *s){
23     while( s[0] == ' ' )
24         strcpy(&s[0], &s[1]);
25
26     while( s[ strlen(s)-1 ] == ' ' )
27         s[ strlen(s)-1 ] = '\0';
28 }
29 void giu1khoangtrang(char *s){
30     for(int i = 0; i < strlen(s); i++){
31
32         if( s[i] == ' ' )
33             while (s[i+1] == ' ')
34                 strcpy(&s[i], &s[i+1]);
35     }
36 }

```

```
37 void daocuoilendau( char *s){
38     char e = s[strlen(s)-1];
39     for(int i = 0; i < strlen(s)-2; i++){
40         char*strcat(char *e, char *s) ;
41     }
42 }
43 void ma(char *s){
44     int m;
45     do {
46         printf ("nhap m = ");
47         scanf ("%d",&m);
48     } while (m > strlen(s));
49     for (int i=0; i<m;i++) {
50         printf ("%c",s[i]);
51     }
52     //kiem tra ma
53     for (int i = 0; i < strlen(s); i++) {
54         if (s[i]=='m' && s[i+1]=='a')
55             s[i]='\0' && s[i+1]='\0';
56         s[strlen(s)-1]='\0';
57     }
58 }
```



```
59 int main (){
60     char s[50];
61     nhap( s);
62     hoasaucham(s);
63     printf (" chuoi sau khi viet hoa sau dau cham la : %s \n",s);
64     thuongsauphay(s);
65     printf ("chuoi ky tu sau khi viet thuong sau dau phay la : %s\n",s);
66     xoa(s);
67     printf ("chuoi ky tu sau khi xoa khoang trang la: %s\n",s);
68     giu1khoangtrang(s);
69     printf ("chuoi ky tu sau khi giu 1 khoang trang la : %s \n",s);
70     daocuoilendau( s);
71     printf ("chuoi ky tu sau khi dao la : %s \n",s);
72     ma(s);
73     printf ("\nchuoi ky tu sau khi xoa ma la : %s \n",s) ;
74 }
```

```
1
2  #include<stdio.h>
3  #include<string.h>
4  // cong tru hai so nguyen lon
5  void swap(char** a, char** b)
6  {
7      char* temp = *a;
8      *a = *b;
9      *b = temp;
10 }
11
12 char* addBigNumber(char* number1, char* number2)
13 {
14     if (strlen(number1) < strlen(number2))
15     {
16         swap(&number1, &number2);
17     }
18
19     size_t number1Len = strlen(number1), number2Len = strlen(number2);
20     char* result = new char[number1Len + 2];
21     memset(result, '0', number1Len);
22     bool remember = false;
23
24     for (int i = 0; i < number2Len; i++)
25     {
26         int temp = number2[number2Len - i - 1] - '0' + number1[number1Len - i - 1] - '0';
27
28         if (remember)
29             temp++;
30
31         remember = temp > 9;
32
33         temp = temp % 10;
34
35         result[number1Len - i] = temp + '0';
36     }
```

```
37
38 for (int i = 0; i < number1Len - number2Len; i++)
39 {
40     int temp = number1[number1Len - number2Len - i - 1] - '0';
41
42     if (remember)
43         temp++;
44
45     remember = temp > 9;
46
47     temp = temp % 10;
48
49     result[number1Len - number2Len - i] = temp + '0';
50 }
51
52 result[number1Len + 1] = '\0';
53
54 if (remember)
55 {
56     result[0] = '1';
57 }
58 else
59 {
60     for (int i = 0; i <= number1Len; i++)
61         result[i] = result[i + 1];
62 }
63
64 return result;
65 }
```

```
66 char* subBigNumber(char* number1, char* number2)
67 {
68     bool sign = false, remember = false;
69
70     if (strlen(number1) < strlen(number2))
71     {
72         sign = true;
73         swap(&number1, &number2);
74     }
75     else if (strlen(number1) == strlen(number2))
76     {
77         for (int i = strlen(number1) - 1; i >= 0; i--)
78         {
79             if (number1[i] < number2[i])
80             {
81                 swap(&number1, &number2);
82                 sign = true;
83             }
84         }
85     }
86     size_t number1Len = strlen(number1);
87     size_t number2Len = strlen(number2);
88     char* result = new char [number1Len + 2];
89     memset(result, '0', number1Len + 2);
90
91     for (int i = 0; i < number2Len; i++)
92     {
93         int temp = number1[number1Len - i - 1] - number2[number2Len - i - 1];
94
95         if (remember)
96             temp--;
97         if (temp < 0)
98         {
99             temp += 10;
```

99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133

```
        temp += 10;
        remember = true;
    }
    else
        remember = false;

    result[number1Len - i] = temp + '0';
}

for (int i = 0; i < number1Len - number2Len; i++)
{
    int temp = number1[number1Len - number2Len - i - 1] - '0';

    if (remember == true)
        temp--;
    if (temp < 0)
    {
        temp += 10;
        remember = true;
    }
    else
        remember = false;

    result[number1Len - number2Len - i] = temp + '0';
}

while (result[0] == '0')
{
    if (result[1] != '0' && sign == true)
    {
        result[0] = '-';
        result[number1Len + 1] = '\\0';
        break;
    }
}
```

```

132     }
133
134     for (int i = 0; i < number1Len; i++)
135     {
136         result[i] = result[i + 1];
137     }
138     result[number1Len] = '\0';
139 }
140
141     return result;
142 }
143
144
145 int main()
146 {
147     char* a = "12345678987654321", *b = "98765432123456789";
148     char* c = addBigNumber(a, b);
149     printf("%s", c);
150     char* d = subBigNumber(a, b);
151     printf("\n%s", d);
152     delete[] c;
153     delete[] d;
154 }

```

Có hai công đoạn khi thực hiện cộng 2 số:

- Tính tổng các chữ số của cả `number1` và `number2` (phần bên phải).
- Tính tổng các chữ số của `number1` với `0` (phần bên trái).

$$\begin{array}{r} + \text{ a } 123456789 \\ \text{ b } 12345 \end{array}$$

```
for (int i = 0; i < number2Len; i++)  
{  
    //...  
}
```

Cần tính tổng các chữ số từ phải qua trái, lấy giá trị các chữ số này bằng cách sử dụng `number1[number1Len - i - 1]` và `number2[number2Len - i - 1]`.

Mỗi giá trị hiện tại đang được lưu trong mảng với kiểu `char`, để lấy được chính xác giá trị kiểu số của các chữ số này, trừ mỗi giá trị cho `48` (`'0'`). Ví dụ: ký tự `'9'` trong kiểu `char` có giá trị là `57`, lấy `57 - 48` được `9`.

Dùng một biến tạm để lưu tổng của 2 giá trị này:

```
int temp = number2[number2Len - i - 1] - '0' + number1[number1Len - i - 1] - '0';
```

Ví dụ: khi cộng `18` và `13`, kết quả là `11` ≥ 10 nên chỉ lấy chữ số hàng đơn vị là `1`, nhớ `1`, tiếp theo, lấy `1 + 1` được `2`, vì có nhớ nên ghi `3`. Kết quả là `31`.

Theo logic đó, tiếp xét giá trị biến `remember`, nếu là `true` thì cộng `temp` lên 1 đơn vị, nếu `temp` có giá trị ≥ 10 , gán `remember = true`, ngược lại ta gán `remember = false`. Để lấy chữ số hàng đơn vị của `temp`, ta lấy `temp` chia phần dư cho 10.


```
if (remember)
    temp++;

remember = temp > 9;

temp = temp % 10;
```

Tổng các chữ số của `number1` và `number2` đã được tính với cách trên, gán giá trị này vào mảng `result`. Tương tự cách làm ở trên, để ghi được ký tự `'9'` vào mảng, cộng số `9` với `48` (`'0'`).

```
*(result + number1Len - i) = temp + '0';
```

Vậy ta có code hoàn chỉnh cho vòng lặp vừa rồi như sau:

```
for (int i = 0; i < number2Len; i++)
{
    int temp = *(b + number2Len - i - 1) - '0' + *(a + number1Len - i - 1) - '0';

    if (remember)
        temp++;

    remember = temp > 9;

    temp = temp % 10;

    result[number1Len - i] = temp + '0';
}
```

Tương tự các bước của công đoạn thứ nhất, số lần lặp của công đoạn này là:

`number1Len - number2Len`.

Giai đoạn này lấy giá trị các ký số của `number1` bằng cách sử dụng: `number1[number1Len - number2Len - i - 1]`, từ đó có giá trị biến `temp` là:

```
int temp = number1[number1Len - number2Len - i - 1] - '0';
```

Tương tự giá trị `result` như sau:

```
result[number1Len - number2Len - i] = temp + '0';
```

Code hoàn chỉnh:

```
for (int i = 0; i < number1Len - number2Len; i++)
{
    int temp = *(number1 + number1Len - number2Len - i - 1) - '0';

    if (remember)
        temp++;

    remember = temp > 9;

    temp = temp % 10;

    result[number1Len - number2Len - i] = temp + '0';
}
```

Để đánh dấu kết thúc chuỗi `result`, gán:

```
result[number1Len + 1] = '\\0';
```

Ví dụ `1 + 9999`, làm lần lượt các bước như trên, được `result` có 4 chữ số, nhưng đáp án là `10000` có 5 ký số. Để làm được việc này, xét giá trị biến `remember`, nếu là `true`: gán giá trị `result[0] = '1'`, ngược lại, dịch tất cả các phần tử của mảng `result` sang bên trái một đơn vị.

```
if (remember)
{
    result[0] = '1';
}
else
{
    for (int i = 0; i <= number1Len; i++)
        result[i] = result[i + 1];
}
```

Cuối cùng, trả về `result` là kết quả của `number1 + number2`.

Kiểm thử với hàm `main`:

```
int main()
{
    char* a = "12345678987654321", *b = "98765432123456789";
    char* c = addBigNumber(a, b);
    printf("%s", c);
    delete[] c;
}
```

```
1  #include<stdio.h>
2  #include<string.h>
3  // dem so lan xuat hien trong xau
4
5  void nhap(char *s){
6      printf("nhap chuoi: ");
7      gets(s);
8  }
9  int dem(char *s, char c){
10     int dem = 0;
11     for (int i=0; i< strlen(s); i++){
12         if (s[i] == c)
13             dem ++;
14     }
15     return dem;
16 }
17 int main (){
18     char s[50];
19     nhap( s);
20     char c; scanf (" %c ",&c);
21     char *p; p = s; gets (p);
22     printf (" so lan xuat hien %c trong xau %s la %d", c,p,dem(s,c));
23 }
24
```

```

1  #include<stdio.h>
2  #include<math.h>
3  #include<string.h>
4  // bai 4 chuong 1
5  void nhap( float *a ,int n, int N){
6      for(int i=0 ; i<n ; i++){
7          for( int j=0 ; j<n ; j++){
8              printf("a[%d][%d]= ",i,j);
9              scanf("%f",a + i*N + j);
10         }
11     }
12 }
13 void xuat(float *a ,int n, int N){
14     for(int i=0 ; i<n ; i++){
15         for(int j=0 ; j<n ; j++){
16             printf("%2.f ", *(a + i*N + j));
17         }
18         printf("\n");
19     }
20 }
21 void phantuan(float *a ,int n, int N){
22     for(int i=0 ; i<n ; i++){
23         for(int j=0 ; j<n ; j++){
24             if ( *(a + i*N + j) <0)
25                 printf (" phan tu am dau tien a[%d][%d] = %1.f \n",i,j,*(a + i*N + j));
26             break;
27         }
28     }
29 }

```

```

30 double tongcheophu(float *a ,int n, int N){
31     double t=0;
32     for(int i=0 ; i<n ; i++){
33         for(int j=0 ; j<n ; j++){
34             if( j == n-1-i)
35                 t+= *(a + i*N + j) ;
36         }
37     } return t;
38 }
39 double max(float *a ,int n, int N){
40     double max = *(a + 0*N + 0);
41     for(int i=0 ; i<n ; i++){
42         for(int j=0 ; j<n ; j++){
43             if(max < *(a + i*N + j))
44                 max= *(a + i*N + j);
45         }
46     }
47     return max;
48 }
49 float tichtungcot(float *a ,int n, int N, float *th){
50     for(int j=0 ; j<n ; j++){
51         th[j] = 1;
52         for(int i=0 ; i<n ; i++){
53             th[j] *= *(a + i*N + j);
54         }
55     }
56     for(int j=0 ; j<n ; j++){
57         printf ( " %.2f ", th[j]);

```

```

58 - double tichmin(float *a ,int n, int N, float *th){
59     //tichtungcot(a ,n,50,th);
60     double min=th[0];
61 -     for(int j=0 ; j<n ; j++){
62 -         for(int j=0 ; j<n ; j++){
63             int min=th[0];
64             if(min>th[j])
65                 min=th[j];
66         }
67     } return min;
68 }
69 - int matrantg(float *a ,int n, int N){
70     int kt1=0, kt2=0;
71 -     for(int i=0 ; i<n ; i++){
72 -         for(int j=0 ; j<n ; j++){
73             if(i<j && *(a + i*N + j)==0)
74                 kt1++;
75             if(j<i && *(a + i*N + j)==0)
76                 kt2++;
77         }
78     }
79 -     if(kt1== (n*(n-1))/2) {
80         return 2;
81     }else
82 -     if(kt2==(n*(n-1))/2) {
83         return 1;
84     }
85     }else {
86         return 0;}
87 }

```



```

88  //- Kiểm tra ma trận có phải là ma phương hay không (tổng các hàng bằng tổng các cột và bằng tổng các phần tử nằm trên 2 đường chéo).
89  void maphuong(float *a ,int n, int N){
90      int kt=1;
91      int cot[n],hang[n], ch=0, ph=0;
92      for(int i=0 ; i<n ; i++){
93          for(int j=0 ; j<n ; j++){
94              cot[j] += *(a + i*N + j);
95              hang[i] += *(a + i*N + j);
96              if(i==j)
97                  ch += *(a + i*N + j);
98              if(j==n-1-i)
99                  ph += *(a + i*N + j);
100          }
101      }
102      for(int i=0;i<n;i++){
103          for(int j=0;j<n;j++){
104              if(hang[i]!=cot[j] || hang[i]!=ch || hang[i]!=ph || cot[j]!=ch || cot[j]!=ph || ch!=ph)
105                  kt=0;
106          }
107      }
108      if(kt==1)
109          printf("\nma tran nay la ma tran ma phuong");
110      else
111          printf("\nma tran nay khong phai la ma tran ma phuong");
112  }
113

```



```

114 int main (){
115     int n ;
116     float a[50][50];
117     float h[50];
118     printf("cap cua ma tran vuong la: ");
119     scanf("%d",&n);
120     nhap( (float*)a , n, 50);
121     xuat( (float*)a , n, 50);
122     phantuan((float*)a , n, 50);
123     printf("tong cac phan tu tren duong cheo phu : %1.f",tongcheophu((float*)a ,n, 50));
124     printf("\ngia tri lon nhat cua ma tran la: %1.f \n", max((float*)a , n, 50));
125     tichtungcot((float*)a , n, 50, h);
126     printf (" %f ",tichmin((float*)a , n, 50,h));
127     if (matrantg((float*)a , n, 50) == 0 )
128         printf (" \nday ko la ma tran tam giac \n");
129     if (matrantg((float*)a , n, 50) == 1)
130         printf (" day la ma tran tam giac");
131     else if (matrantg((float*)a , n, 50) == 2 )
132         printf(" ma tran tam giac duoi");
133
134     maphuong((float*)a,n,50);
135 }

```

```

1  #include<stdio.h>
2  #include<math.h>
3  void nhap(int n,int m, float a[100][100])
4  {
5      for(int i=1;i<=n;i++)
6          for(int j=1;j<=m;j++)
7          {
8              printf("a[%d][%d] ",i,j);
9              scanf("%f",&a[i][j]);
10         }
11     }
12     void xuat(int n,int m,float a[100][100])
13     {
14         for(int i=1;i<=n;i++)
15         {
16             for(int j=1;j<=m;j++)
17                 printf("%.2f ",a[i][j]);
18             printf("\n");
19         }
20     }
21     float max(int n,int m,float a[100][100])
22     {
23         float max=a[1][1];
24         for(int i=1;i<=n;i++)
25             for(int j=1;j<=m;j++)
26                 if(max<a[i][j]) max=a[i][j];
27         return max;
28     }

```

```

29 float tinhTong(int n,int m,int u,int v,float a[100][100])
30 {
31     float t=0;
32     if(m!=n) printf("Ko co duong cheo chinh\n ");
33     else if(u==v) printf("Diem nam tren duong cheo chinh\n");
34     else
35     {
36         if(u>v)
37             for(int i=1;i<=n-(u-v);i++)
38                 t=t+a[u-v+i][i];
39         if(u<v)
40             for(int i=1;i<=n-(v-u);i++)
41                 t=t+a[i][v-u+i];
42         printf("Tong phan tu nam tren duong thang // dxc qua diem a[%d][%d] = %.2f",u,v,t);
43     }
44 }
45 void dinhthuc(int n,int m,float a);
46 int main()
47 {
48     int i,j,m,n,u,v;
49     float a[100][100];
50     printf("Nhap hang :");
51     scanf("%d",&n);
52     printf("Nhap cot :");
53     scanf("%d",&m);
54     nhap(n,m,a);
55     xuat(n,m,a);
56     if(max(n,m,a)>0)
57         printf("Gia tri duong lon nhat: %.2f\n",max(n,m,a));
58     printf("Nhap u:");
59     scanf("%d",&u);
60     printf("Nhap v:");
61     scanf("%d",&v);
62     tinhTong(n,m,u,v,a);
63 }

```

```

1  #include<stdio.h>
2  #include<math.h>
3  #include<string.h>
4  // xoay 90 vs doi xung
5  void nhap( float *a ,int n, int N){
6      for(int i=0 ; i<n ; i++){
7          for( int j=0 ; j<n ; j++){
8              printf("a[%d][%d]= ",i,j);
9              scanf("%f",a + i*N + j);
10         }
11     }
12 }
13 void xuat(float *a ,int n, int N){
14     for(int i=0 ; i<n ; i++){
15         for(int j=0 ; j<n ; j++){
16             printf("%2.f ", *(a + i*N + j));
17         }
18         printf("\n");
19     }
20 }
21 void xoay( float *a ,int n, int N, float *x){
22     for(int i=0;i<n;i++){
23         int k=n-1;
24         for(int j=0 ; j<n ; j++) {
25             *(x + i*N + j) = *(a + k*N + i);
26             k--;
27         }
28     }
29 }

```

```

30 int kiemtradoixung(float *a,int n,int N){
31     int kt=1;
32     for(int i=0;i<n;i++){
33         for(int j=0;j<n;j++){
34             if(*(a + j*N + i) != *(a + j*N + i))
35                 kt = 0; break;
36         }
37     } return kt;
38 }
39 int main (){
40     int n ;
41     float a[50][50];
42     float x[50][50];
43     printf("cap cua ma tran vuong la: ");
44     scanf("%d",&n);
45     nhap( (float*)a , n, 50);
46     xuat( (float*)a , n, 50);
47     xoay( (float *)a , n , 50, (float *)x);
48     printf(" ma tran vuong la: \n");
49     xuat( (float*)x , n, 50);
50     kiemtradoixung((float*)a , n, 50);
51     if(kiemtradoixung((float*)a , n, 50) ==0)
52         printf("KHONG DOI XUNG\n");
53     else
54         printf("DOI XUNG\n");
55 }

```