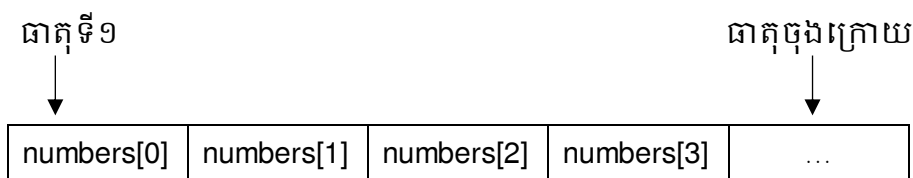


១. សេចក្តីផ្តើមនៃ array

array គឺជាសំណុំអញ្ញាតដែលមានប្រភេទទិន្នន័យដូចគ្នា។ គេអាចផ្ទុកសំណុំតម្លៃដែលមានប្រភេទទិន្នន័យដូចគ្នានៅក្នុង array មួយ។ លក្ខណៈរបស់ array មានដូចខាងក្រោម៖

- ប្រភេទរបស់ array មួយអាចជាប្រភេទណាមួយ (int, float, char, double, ...)
- ទំហំ array ត្រូវតែជាតម្លៃថេរ (មិនអាចជាលេខទសភាគបានឡើយ)
- គ្រប់ array ទាំងអស់មានទីតាំង memory ជាប់ៗគ្នាហើយត្រូវបានប្រើសម្រាប់ផ្ទុកធាតុនៃ array ក្នុង memory។ អាស័យដ្ឋានទាបបំផុតត្រូវនឹងធាតុទីមួយនិងអាស័យដ្ឋានខ្ពស់បំផុតត្រូវនឹងធាតុចុងក្រោយ។

ដើម្បីជំនួសការប្រកាសអញ្ញាតដោយឡែកពីគ្នា ដូចជា number0, number1, ... និង number99 គេអាចប្រកាសអញ្ញាត array មួយ ដូចជា numbers ដោយប្រើ numbers[0], numbers[1], ... និង numbers[99] សម្រាប់តាងឲ្យអញ្ញាតដោយឡែកពីគ្នានោះ។ គេអាចចូលប្រើធាតុណាមួយរបស់ array តាមរយៈការបញ្ជាក់លេខ index។



គេចែក array ជាពីរប្រភេទធំៗ គឺ array មួយវិមាត្រ និង array ច្រើនវិមាត្រ (រួមមាន array ពីរ វិមាត្រ, array បីវិមាត្រ, array បួនវិមាត្រ, ...)។ គេអាចសម្គាល់បាន array មួយវិមាត្រគឺមាន index តែមួយគត់ដាក់នៅក្នុងឃ្លាប [] ។ រីឯ array ច្រើនវិមាត្រគឺមាន index ចាប់ពីពីរឡើងទៅដាក់នៅក្នុងឃ្លាប []។

២. ការប្រកាស array មួយវិមាត្រ

ដើម្បីប្រកាស array មួយក្នុងភាសា C អ្នកសរសេរកម្មវិធី ត្រូវបញ្ជាក់ប្រភេទធាតុ និងចំនួនធាតុតាមតម្រូវការរបស់ array ។

ទម្រង់ទូទៅរបស់វា ៖

```
type arrayName [ arraySize ];
```

array នេះជា array មួយវិមាត្រ។ arraySize ត្រូវតែជាចំនួនគត់ថេរមានតម្លៃជំនាន់សូន្យ ហើយ type ជាប្រភេទទិន្នន័យណាមួយរបស់ភាសា C។ ឧទាហរណ៍ ដើម្បីប្រកាស array ដែលមានទំហំ ១០ធាតុ ជាប្រភេទ double ដោយមានឈ្មោះ balance នោះ គេសរសេរ៖

```
double balance[10];
```

៣. ការកំណត់តម្លៃដំបូង array មួយ

គេអាចកំណត់តម្លៃដំបូងទៅឲ្យ array មួយតាមវិធីពីរយ៉ាង។

ឧទាហរណ៍ ៖ `int a[5];`

```
a[0] = 10;
```

```
a[1] = 20;
```

```
a[2] = 30;
```

```
a[3] = 40;
```

```
a[4] = 50;
```

ឬ `int b[5] = {10, 20, 30, 40, 50};`

គេមិនដាក់លេខ index ក្នុង array b ក៏បាន ពោលគឺគេអាចសរសេរជា៖

```
int b[ ] = {10, 20, 30, 40, 50};
```

ចំនួនតម្លៃក្នុងចន្លោះសញ្ញា { } មិនអាចច្រើនជាងចំនួនធាតុដែល array បានប្រកាសជាក់ក្នុងសញ្ញា [] ឡើយ។

គ្រប់ array ទាំងអស់តែងមាន index សូន្យនៅត្រង់ធាតុទីមួយ ហើយធាតុចុងក្រោយបង្អស់មាន index ស្មើនឹងទំហំ array ដកនឹងមួយ។ គំនូសតាងខាងក្រោមបង្ហាញពីធាតុនៃ array និង index របស់វា។

index:	0	1	2	3	4
b	10	20	30	40	50

៤. ការចូលប្រើធាតុមួយរបស់ array

គេអាចចូលប្រើធាតុណាមួយរបស់ array ដោយបញ្ជាក់លេខ index។ ក៏ប៉ុន្តែ គេមិនអាចដាក់លេខ index លើសពីទំហំរបស់ array ឡើយ។

ឧទាហរណ៍ ៖ `int x = b[4]; // x = 50`

បើយើងចង់ធ្វើផលបូកតម្លៃទាំងអស់របស់ array b ផ្ទុកក្នុងអញ្ញាត sum នោះ គេអាចសរសេរ៖

```
int sum = b[0] + b[1] + b[2] + b[3] + b[4];
```

ឬ

```
for (int i=0; i < 5; i++)
```

```
    sum = sum + b[ i ];
```

៥. ការបញ្ចូលតម្លៃទៅឱ្យធាតុ array តាមរយៈ Keyboard

ជាធម្មតា គេកំណត់តម្លៃទៅឱ្យធាតុនីមួយៗរបស់ array កាលណាគេដឹងចំនួនធាតុ array ពិតប្រាកដ។ គេមិនអាចកំណត់តម្លៃឱ្យធាតុនៃ array ដែលធំជាងទំហំ array បានឡើយ។ កម្មវិធីខាងក្រោមនេះបង្ហាញពីការបញ្ចូលតម្លៃតាមរយៈ keyboard ទៅដាក់ក្នុង array មួយ រួចហើយធ្វើផលបូកតម្លៃទាំងអស់។

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

```
#include <conio.h>
```

```
void main(){
```

```
    int a[10]; // ការប្រកាស array a មានប្រវែង 10 ធាតុជាប្រភេទ int
```

```
    int n, sum=0; // ការប្រកាសនិងកំណត់តម្លៃអញ្ញាត sum = 0, និង n ជាប្រភេទ int
```

```
    printf("How many values do you want to input ?\n");
```

```
    // បង្ហាញឃ្លានេះនៅលើអេក្រង់
```

```
    scanf("%d", &n); // ទទួលតម្លៃពី keyboard និងផ្ទុកឱ្យអញ្ញាត n
```

```
    if (n > 10){ // បើ n > 10 នោះ
```

```
        printf("It is greater than array size.\n");
```

```
        // បង្ហាញឃ្លានេះនៅលើអេក្រង់
```

```
        getch();
```

```
        return;
```

```
    }
```

```

for (int i=0; i<n; i++){ // បើ i < n នោះ

    printf("Input a value : "); // បង្ហាញឃ្លានេះនៅលើអេក្រង់
    scanf("%d", &a[i]); // ទទួលតម្លៃពី keyboard និងផ្ទុកឲ្យអញ្ញាត a[i]
}
for (int i=0; i<n; i++)
    sum += a[i]; // បូកបន្ថែមតម្លៃ a[i] ទៅឲ្យអញ្ញាត sum
printf("sum of array is %d\n", sum); // បង្ហាញតម្លៃ sum មកលើអេក្រង់
getch();
}

```

៦. ការប្រើ array ប្រភេទ char ជា string

តាមធម្មតា គេប្រកាស array ជាប្រភេទ char ដូចទៅនឹងប្រភេទទិន្នន័យផ្សេងទៀតដែរ។ យើងត្រូវចាំថា string នៃ array ប្រភេទ char មួយវិមាត្រត្រូវបញ្ចប់ដោយតួអក្សរ null character ('\0')។

ឧទាហរណ៍៖

```

char greeting[8] = { 'W', 'e', 'l', 'c', 'o', 'm', 'e', '\0'};
ឬ char greeting[ ] = { 'W', 'e', 'l', 'c', 'o', 'm', 'e', '\0'};

#include <stdio.h>
#include <conio.h>
void main(){
    char greeting[8] = { 'W', 'e', 'l', 'c' , 'o', 'm', 'e'};
    for (int i=0; i<8; i++)
        printf("%c", greeting[i]); // បង្ហាញតម្លៃរបស់ greeting[i] មកលើអេក្រង់
    getch();
}

```

ឬ គេអាចសរសេរម៉្យាងទៀត

```

#include <stdio.h>
#include <conio.h>
void main(){

```

```

char greeting[] = {'W', 'e', 'l', 'c' , 'o', 'm', 'e',
                  '\0'};

printf("%s\n", greeting); // បង្ហាញតម្លៃរបស់ greeting មកលើអេក្រង់
getch();
}

```

កំណត់សម្គាល់៖ គេអាចប្រើអនុគមន៍ដែលមានស្រាប់របស់ cstring.h ដូចជា

- strcpy(s1, s2) : ចម្លង string s2 ទៅឲ្យ string s1 ។
- strlen(s1) : ឲ្យតម្លៃជាប្រវែង string s1 ។
- strcat(s1, s2) : ភ្ជាប់តម្លៃ string s2 ទៅខាងចុងនៃ string s1 ។
- strcmp(s1, s2) : ឲ្យតម្លៃ សូន្យ បើ string s1 និង string s2 ដូចគ្នា ហើយឲ្យតម្លៃ តូចជាងសូន្យ

កាលណា string s1 < s2 និងឲ្យតម្លៃ ធំជាងសូន្យ កាលណា string s1 > s2 ។

ឧទាហរណ៍៖ កម្មវិធីខាងក្រោមនេះបង្ហាញពីការប្រើអនុគមន៍ string មួយចំនួនដូចបានរៀប រាប់ ខាងលើនេះ។

```

#include <stdio.h>
#include <cstring.h>
#include <conio.h>

void main(){
    char greeting[8] = {'W', 'e', 'l', 'c' , 'o', 'm', 'e',
                       '\0'};

    char ch[8]; // ការប្រកាសអញ្ញាត array ch ចំនួន 8 ធាតុ
    strcpy(ch, greeting); // ចម្លងតម្លៃរបស់ greeting ទៅឲ្យ ch
    printf("%s\n", ch); // ទទួលតម្លៃពី keyboard និងផ្ទុកឲ្យអញ្ញាត ch
    printf("%d\n", strcmp(ch, greeting));
    printf("%d\n", strlen(ch));

    strcat(ch, greeting); // ភ្ជាប់តម្លៃរបស់ greeting ទៅនឹងអញ្ញាត ch
    printf("%s\n", ch); // បង្ហាញតម្លៃរបស់ ch មកលើអេក្រង់
    getch();
}

```

៧. ការប្រើ array ជាមួយនឹងអនុគមន៍

វាមានពីរវិធីសម្រាប់ប្រកាសអនុគមន៍មួយដោយមានប៉ារ៉ាម៉ែត្ររបស់វាជា array ។

ទម្រង់ទូទៅរបស់វា គឺ៖

```
return_type functionName(data_type arrayName[ ], int arraySize);
```

ឬ

```
return_type functionName(data_type* arrayName, int arraySize);
```

ឧទាហរណ៍៖ អនុគមន៍ខាងក្រោមឈ្មោះ sorting មានប៉ារ៉ាម៉ែត្រពីរគឺទី១ជា array ដែលផ្ទុកតម្លៃ ហើយបញ្ជូនចូល ទៅក្នុង array នោះ និងប៉ារ៉ាម៉ែត្រទី២ ជាប្រភេទ int តាងឲ្យចំនួនធាតុរបស់ array ។

```
#include <stdio.h>
#include <conio.h>
void sorting(int x[], int n){
    int i, j, temp ;    // ការប្រកាសអញ្ញាត i, j, temp ជាប្រភេទ int
    for(i=1; i<= n-1; i++){
        for(j=0; j< n-i; j++){
            if(x[j] > x[j+1]){    // បើ x[j]>x[j+1] នោះ
                temp = x[j];    // កំណត់តម្លៃរបស់ x[j] ទៅឲ្យអញ្ញាត temp
                x[j] = x[j+1];    // កំណត់តម្លៃរបស់ x[j+1] ទៅឲ្យអញ្ញាត x[j]
                x[j+1] = temp;    // កំណត់តម្លៃរបស់ x[j+1] ទៅឲ្យអញ្ញាត temp
            }
        }
    }
    for(i=0; i<n; i++){
        printf("%d\t", x[i]);    // បង្ហាញតម្លៃរបស់ x[i] មកលើអេក្រង់
    }
}
void main(){
    int a[] = {34, 10, 84, 21, 8, 76, 56 };
    sorting(a, 7);    // ការហៅអនុគមន៍ sorting() មកប្រើ
    getch();
}
```


កម្មវិធីខាងក្រោមបង្ហាញពីរបៀបបញ្ចូលតម្លៃពី keyboard ទៅផ្ទុកក្នុង array ពីរវិមាត្រដែលមានចំនួន 12 ធាតុ។

```
#include <stdio.h>
#include <conio.h>
void main() {
    int a[3][4]; // ការប្រកាសអញ្ញាត array a[ ][ ] ជាប្រភេទ int
    int i,j; // ការប្រកាសអញ្ញាត i, និង j ជាប្រភេទ int
    printf("Enter 12 array elements\n");
    for(i=0; i < 3; i++){
        for(j=0; j < 4; j++){
            printf("Enter a value : ");
            scanf("%d", &a[i][j]); // ទទួលតម្លៃពី keyboard និងផ្ទុកឱ្យអញ្ញាត a[i][j]
        }
    }
    printf("You have entered all these values.\n");
    for(i=0; i < 3; i++){
        for(j=0; j < 4; j++){
            printf("%d ", a[i][j]); // បង្ហាញតម្លៃរបស់ a[i][j] មកលើអេក្រង់
        }
    }
    getch();
}
```

កម្មវិធីខាងក្រោមបង្ហាញលទ្ធផលជារាងត្រីកោណ Pascal កាលណាគេបញ្ចូលតម្លៃពី keyboard នូវចំនួន rows ដែលត្រូវបង្ហាញ។

```
#include <stdio.h>
#include <conio.h>
#define MAX 16
void main() {
    int n;
    int a[MAX][MAX];
```



```

printf("Display Pascal Triangle Shape.\n");
printf("Enter a value (number of rows <= 15) : ");
scanf("%d", &n); // ទទួលតម្លៃពី keyboard និងផ្ទុកឲ្យអញ្ញាត n
if ( n <= 0 ) { // បើ n <= 0 នោះ
    printf("Invalid value."); // បង្ហាញឃ្លានេះមកលើអេក្រង់
    getch();
    return;
}
if ( n >= 15 ) { // បើ n >= 15 នោះ
    printf("It is too big to display on screen.");
    // បង្ហាញឃ្លានេះមកលើអេក្រង់
    getch();
    return;
}
for (int i=0; i<=n; i++)
    a[i][0] = a[i][i] = 1;
for (int i=2; i<=n; i++)
    for (int j=1; j<i; j++)
        a[i][j] = a[i-1][j-1] + a[i-1][j];
for (int i=0; i<=n; i++){
    printf("\n");
    for (int j=0; j<=i; j++)
        printf("%d\t", a[i][j]);
}
getch();
}

```

ចំពោះករណីការប្រកាស array បីវិមាត្រមានលក្ខណៈស្រដៀងនឹង array ពីរវិមាត្រដែរ ពោលគឺវាមានសញ្ញាប្លុក [] ចំនួនបីនៅពីក្រោយឈ្មោះ array។

ឧទាហរណ៍៖ `int a[2][3][4] = { {1, 2, 3, 4},`
`{5, 6, 7, 8},`

{9, 10, 11, 12}},
 {{13, 14, 15, 16},
 {17, 18, 19, 20},
 {21, 22, 23, 24}} } };

a

		[0]	[1]	[2]	[3]
	[0]	1	2	3	4
	[0] [1]	5	6	7	8
	[2]	9	10	11	12

		[0]	[1]	[2]	[3]
	[0]	13	14	15	16
[1]	[1]	17	18	19	20
	[2]	21	22	23	24

សំណួរនិងលំហាត់

១. អ្វីទៅ array ? ចូរនិយាយពីសារៈសំខាន់នៃការប្រើ array និង គុណវិបត្តិនៃការប្រើ array។
២. ចូរសរសេរអនុគមន៍មួយដែលអាចឲ្យគេបញ្ចូលតម្លៃតាមប៉ារ៉ាម៉ែត្រជា array មួយវិមាត្រចំនួន n ធាតុ ដើម្បីឲ្យគេអាចរកតម្លៃអតិបរមា និង អប្បបរមាក្នុង array នោះ។
៣. ចូរសរសេរអនុគមន៍មួយដែលអាចឲ្យគេបញ្ចូលតម្លៃតាមប៉ារ៉ាម៉ែត្រជា array មួយវិមាត្រចំនួន n ធាតុ ដើម្បីឲ្យគេអាចរាប់ចំនួនលេខសេស និង លេខគូក្នុង array នោះ។
៤. ចូរសរសេរ code ដើម្បីឲ្យគេអាចបញ្ចូលតម្លៃពី keyboard ទៅផ្ទុកក្នុង array មួយវិមាត្រ ហើយ អាចឲ្យគេរកតម្លៃមួយ ក្នុង array នោះថា តើមានតម្លៃនោះឬក៏អត់។
៥. តើ code ខាងក្រោមនេះឲ្យគេទទួលបានលទ្ធផលអ្វីនៅលើអេក្រង់? ចូរពន្យល់។

```
#include <stdio.h>
#include <conio.h>
void main(){
    int a[5] = {5, 1, 15, 20, 25};
    int i, j, m;
    i = ++a[1];
    j = a[1]++;
    m = a[i++];
    printf("%d, %d, %d", i, j, m);
    getch();
}
```

៦. តើ code ខាងក្រោមនេះឲ្យគេទទួលបានលទ្ធផលអ្វីនៅលើអេក្រង់? ចូរពន្យល់។

```
#include <stdio.h>
#include <conio.h>
void main(){
    int list1[5]={1,3,5,7,9}; //initialize list1 array
    int list2[5];
    int i;
    for(i=0; i<5; i++)
        list2[i] = list1[i]+1; //assign values to list2
    for(i=0; i<5; i++) //output list2
```

```
        printf("%d\t", list2[i]);  
    getch();  
}
```