



ບົດທີ 6

ອາເຣຍ (Array)





ເນື້ອໃນຫຍໍ້

- I. ຄວາມໝາຍຂອງອາເຣຍ (array).
- II. ອາເຣຍ 1 ມິຕິ.
- III. ອາເຣຍ 2 ມິຕິ.
- IV. ອາເຣຍຫຼາຍມິຕິ.



I. ຄວາມໝາຍຂອງອາເຣຍ (array)

array ເປັນຊະນິດຂໍ້ມູນປະເພດໜຶ່ງທີ່ນຳຊະນິດຂໍ້ມູນພື້ນຖານຢ່າງຊະນິດຂໍ້ມູນແບບຕົວອັກສອນ (char), ຊະນິດຂໍ້ມູນແບບຕົວເລກຈຳນວນເຕັມ (int), ຊະນິດຂໍ້ມູນແບບຕົວເລກຈຳນວນຈິງ (float) ມາປະຍຸກເປັນຂໍ້ມູນແບບ array, ເຊິ່ງໂຄງສ້າງຂໍ້ມູນຂອງອາເຣຍຈະເກັບຂໍ້ມູນຕາງຈາກຊະນິດຂໍ້ມູນພື້ນຖານທົ່ວໄປຄື: ຖາປະກາດຕົວປຽນໜຶ່ງຂຶ້ນໄປໃຊງງານ ໂດຍກຳນົດຊະນິດຂໍ້ມູນນີ້ເປັນຊະນິດຂໍ້ມູນພື້ນຖານແລວ ຕົວປຽນນີ້ສາມາດເກັບຄ່າໄດພຽງໜຶ່ງຄ່າເທົ່ານັ້ນ, ແຕ່ຖາປະກາດຕົວປຽນທີ່ມີຊະນິດຂໍ້ມູນແບບ array ນີ້ຂຶ້ນມາໃຊ້ ພາຍໃນໜຶ່ງຕົວປຽນສາມາດມີຄ່າຫຼາຍກວ່າໜຶ່ງຄ່າຂຶ້ນໄປ, ເຊິ່ງຈະເກັບຄ່າໄດຫຼາຍເທົ່າໃດນັ້ນແມ່ນຂຶ້ນກັບຂະໜາດຂອງ array ທີ່ເຮົາໄດ້ກຳນົດໄວ.



I. ຄວາມໝາຍຂອງອາເຣຍ (array)

ຕົວຢ່າງ: ການຂຽນໂປຣແກຣມເພື່ອເກົ້າຕົວເລກ, ເຊິ່ງເຮົາມີເລກໃຫ້ເດົາແນວ 0-10.

```
main(){
    int num0,num1,num2, num3, num4,
    num5, num6, num7, num8, num9, num10;
    printf("\tEnter the guess number:");
    scanf("\t%d",&num0);
    scanf("\t%d",&num1);
    scanf("\t%d",&num2);
    scanf("\t%d",&num3);
    scanf("\t%d",&num4);
    scanf("\t%d",&num5);
    scanf("\t%d",&num6);
    scanf("\t%d",&num7);
    scanf("\t%d",&num8);
    scanf("\t%d",&num9);
    scanf("\t%d",&num10);
}
```

ແບບພື້ນຖານ

```
main(){
    int num[10];
    for(i=0;i<=10;i++){
        printf("\tEnter the guess number:");
        scanf("\t%d\n",&num[i]);
    }
}
```

ແບບອາເຣຍ



II. ອາເຣຍ 1 ມິຕິ

ຮູບແບບຂອງອາເຣຍ 1 ມິຕິ:

ຊະນິດຂໍ້ມູນ ຊື່ຕົວປຽນ[ຂະໜາດຂອງອາເຣຍ];

int a [2];

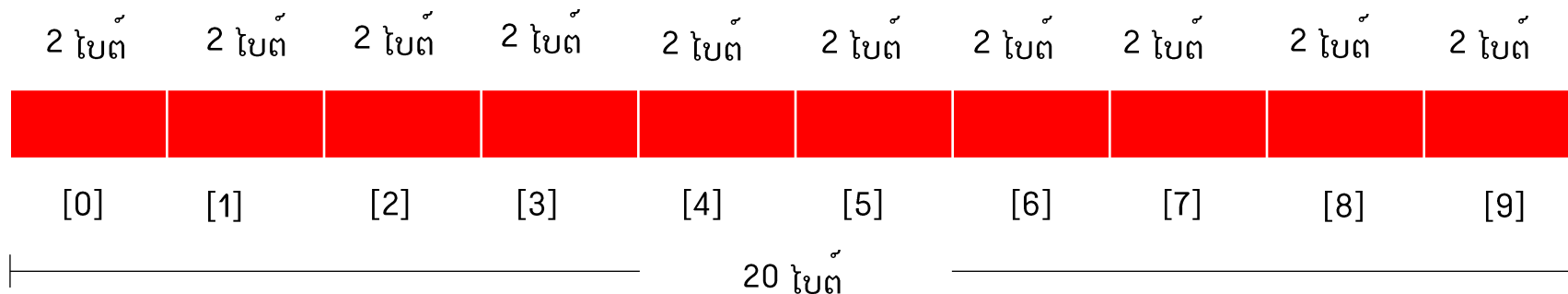
float b [3];

char c[6];



II. ອາເຣຍ 1 ມິຕິ

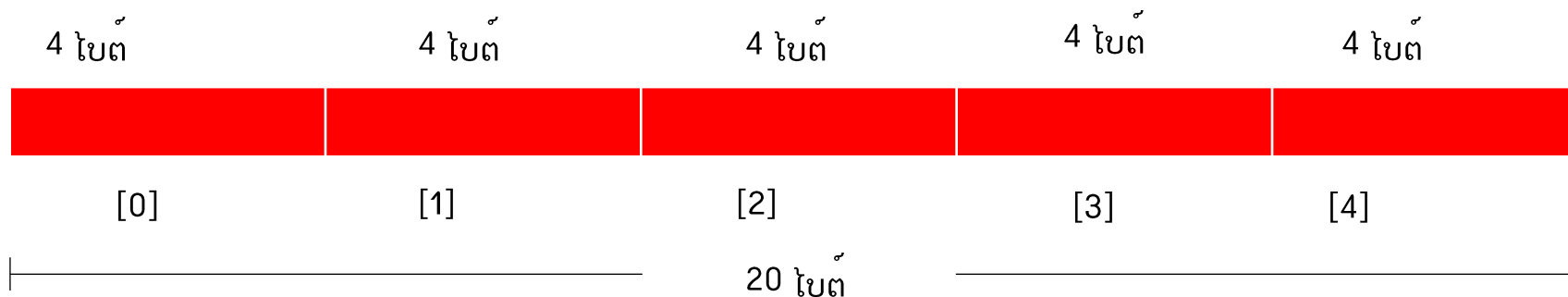
`int a[10];` ປະກາດຕົວປຽນ `a` ເປັນອາເຣຍຊະນິດຂໍ້ມູນຈຳນວນເຕັມ ມີຂະໜາດ 10 ຊອງ ໂດຍຊະນິດຂໍ້ມູນແບບຈຳນວນເຕັມມີ 2 ໄບຕ ແລະ ກຳນົດເປັນຊະນິດຂໍ້ມູນແບບ ອາເຣຍຂະໜາດ 10 ຊອງດັ່ງນັ້ນຕົວປຽນຈຶ່ງມີຂະໜາດ 20 ໄບຕ.





II. ອາເຣຍ 1 ມິຕິ

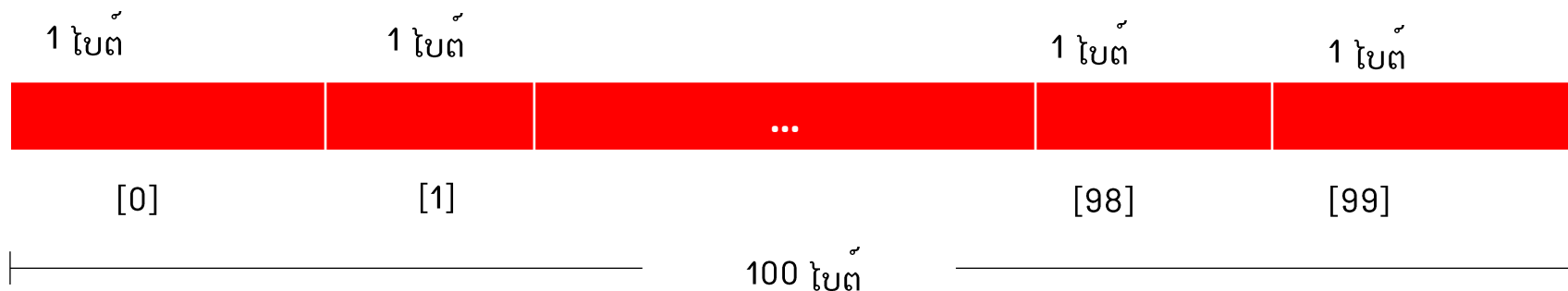
`float b[5]` ປະກາດຕົວປຽນ `b` ເປັນອາເຣຍຊະນິດຂໍ້ມູນຈຳນວນຈິງ ມີຂະໜາດເທົ່າ 5 ຊອງ ໂດຍມີຊະນິດຂໍ້ມູນແບບເລກຈຳນວນຈິງປະເພດ `float` ມີຂະໜາດ 4 ໂບດ ແລະ ກຳນົດເປັນອາເຣຍຂະໜາດ 5 ຊອງ ດັ່ງນັ້ນຕົວປຽນ `b` ຈຶ່ງມີຂະໜາດ 20 ໂບດ.





II. ອາເຣຍ 1 ມິຕິ

`char c[100]` ປະກາດຕົວປຽນ `c` ເປັນອາເຣຍຊະນິດຂໍ້ມູນຕົວອັກສອນ ທີ່ ເອີ້ນອີກຢ່າງໜຶ່ງ ວາແມ່ນ `string` ມີຂະໜາດເທົ່າ 100 ຊອງ ໂດຍມີຊະນິດຂໍ້ມູນແບບປະເພດ `char` ມີຂະໜາດ 1 ໄບຕ໌ ແລະ ກຳນົດເປັນອາເຣຍຂະໜາດ 100 ຊອງ ດັ່ງນັ້ນຕົວປຽນ `c` ຈຶ່ງມີຂະໜາດ 100 ໄບຕ໌.





II. ອາເຣຍ 1 ມິຕິ

ໝາຍເຫດ: ການປະກາດຕົວປຽນແບບອາເຣຍຂຶ້ນໃຊງານຈະຕ້ອງທຳການກຳນົດຂະໜາດຂອງອາເຣຍທີ່ແນ່ນອນວ່າຕ້ອງການອາເຣຍຂະໜາດຈັກຊອງ ແລະ ຂໍ້ມູນທີ່ຕ້ອງເກັບໃນອາເຣຍນັ້ນຕ້ອງເປັນຂໍ້ມູນດຽວກັນຄື: ຖ້າເປັນຊະນິດຂໍ້ມູນແບບຕົວອັກສອນກໍຕ້ອງເປັນຊະນິດຂໍ້ມູນແບບຕົວອັກສອນຄືກັນ, ເປັນຊະນິດຂໍ້ມູນແບບຕົວເລກຈຳນວນເຕັມກໍຕ້ອງເປັນຊະນິດຂໍ້ມູນແບບຕົວເລກຈຳນວນເຕັມຄືກັນ ແລະ ຖ້າເປັນຊະນິດຂໍ້ມູນແບບຕົວເລກຈຳນວນຈົງກໍຕ້ອງເປັນຊະນິດຂໍ້ມູນທີ່ເປັນຕົວເລກຈຳນວນຈົງ.



II. ອາເຣຍ 1 ມິຕິ

ການກຳນົດຄ່າເລີ່ມຕົ້ນໃຫ້ກັບອາເຣຍ 1 ມິຕິ: ການກຳນົດຄ່າເລີ່ມຕົ້ນໃຫ້ກັບອາເຣຍສາມາດທຳໄດ້ ໂດຍທຳການກຳນົດຄ່າເລີ່ມຕົ້ນໃຫ້ກັບຕົວປຽນໄປພ້ອມໆກັບການປະກາດຕົວປຽນຂຶ້ນໃຊ້ງານ ໂດຍຄ່າທີ່ຕ້ອງກຳນົດໃຫ້ກັບອາເຣຍຈະຕ້ອງລະບຸໄວ້ພາຍໃນເຄື່ອງໝາຍ { } ແລະ ຄ່າທີ່ຕ້ອງການກຳນົດໃຫ້ກັບອາເຣຍມີຫຼາຍກວ່າໜຶ່ງຄ່າ ໃຫ້ແຍກແຕ່ລະຄ່າອອກຈາກກັນດ້ວຍເຄື່ອງໝາຍ (,) comma.

ຕົວຢ່າງ:

char A[3] = {'a' , 'b'};

int b[5] = {1,2,3};

char A[3] = {'a' , 'b'} ມີຄວາມໝາຍ char A[3]="ab"

int a[]={1,2,3}; ມີຄວາມໝາຍວ່າ
int a[3]={1,2,3};

A	a	b	0		
	[0]	[1]	[2]		
b	1	2	3	0	0
	[0]	[1]	[2]	[3]	[4]



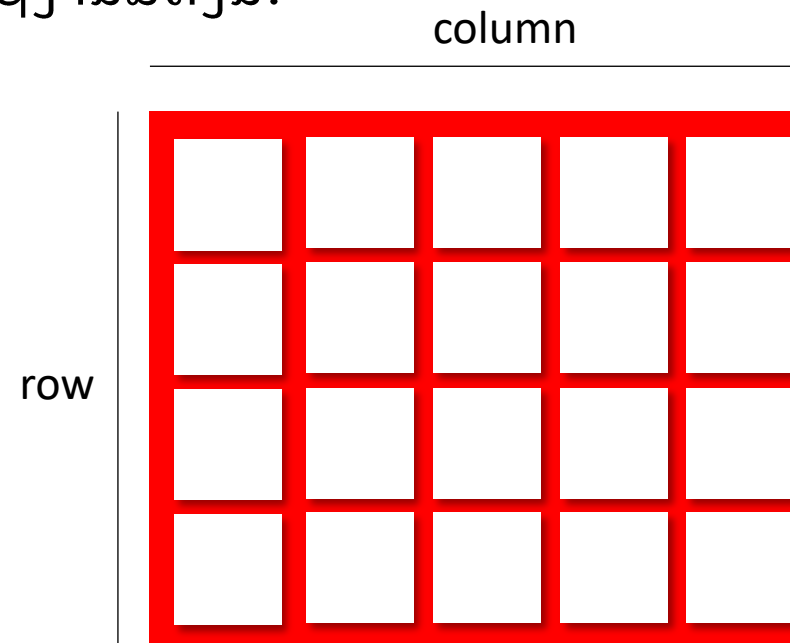
III. ອາເຣຍ 2 ມິຕິ

Array 2 ມິຕິແມ່ນໄດ້ມາຈາກ array 1 ມິຕິ ເຮົາເອີ້ນ array 2 ມິຕິວ່າ array of array, ເຊິ່ງເຮົາມີ array ມິຕິທີ 1 ເປັນ row (ແຖວ), ມິຕິທີ 2 ເປັນ column (ຖັນ). ການປະກາດ array 2 ມິຕິຂຶ້ນໃຊງງານມີດັ່ງນີ້:

ຊະນິດຂໍ້ມູນ ຊື່ຕົວປຽນ [ແຖວ][ຖັນ];

Datatype variable [row][column];

Int num[5][4]





III. ອາເຣຍ 2 ມິຕິ

ການປະກາດຄາເລີ່ມຕົ້ນໃຫ້ກັບ array 2 ມິຕິ:

```
int num [3][4]={ 1,2,3,4,5,6,7,8,9,10,11,12 };
```

```
int num [3][4]={ {1,2,3,4}, {5,6,7,8}, {9,10,11,12} };
```

column

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

row



III. ອາເຣຍ 2 ມິຕິ

ຕົວຢ່າງການນຳໃຊ້ array 2 ມິຕິ:

example1array2.c

```
1  #include<stdio.h>
2  #include<stdlib.h>
3
4  main(){
5      printf("\t=== example 1 array 2 ===\n\n\n");
6      int num [3][4]={ {1,2,3,4},{5,6,7,8},{9,10,11,12} };
7      printf("\t\t%d\n\n\n",num[1][2]);
8      printf("\t=== example 1 array 2 ===\n");
9  }
```



IV. ອາເຣຍຫຼາຍມິຕິ

ອາເຣຍຫຼາຍມິຕິໝາຍເຖິງອາເຣຍຕັ້ງແຕ່ 3 ມິຕິຂຶ້ນໄປ. ຕົວຢ່າງອາເຣຍ 3 ມິຕິ ເຮົາຈະໄດ້ມິຕິທີ 1 ເປັນຈຳນວນແຜ່ນຂອງອາເຣຍ, ມິຕິທີ 2 ເປັນ row, ມິຕິທີ 3 ເປັນ column ຕົວຢ່າງປະກາດ array 3 ມິຕິຂຶ້ນມາໃຊງງານດັ່ງຕໍ່ໄປນີ້:

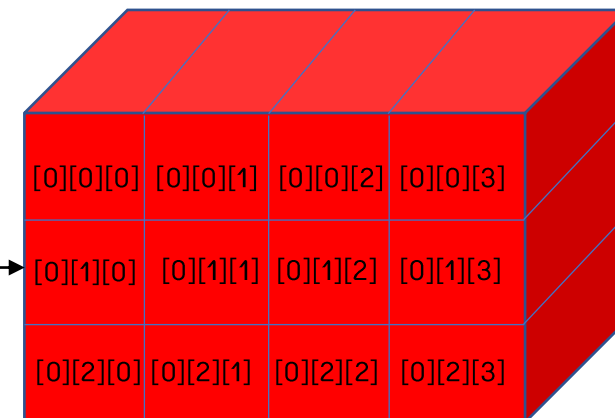
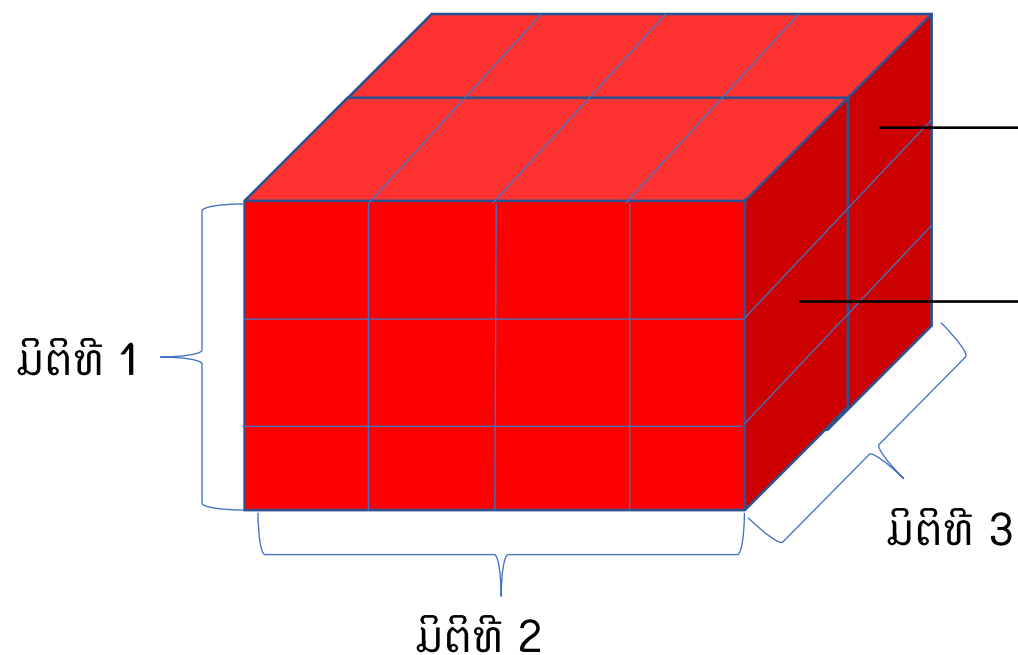
```
int num[2][3][4];
```



IV. ອາເຣຍຫຼາຍມິຕິ

ຕົວຢ່າງປະກາດ array 3 ມິຕິຂຶ້ນມາໃຊງງານດັ່ງຕໍ່ໄປນີ້:

```
int num[2][3][4];
```





IV. ອາເຣຍຫຼາຍມິຕິ

ຕົວຢ່າງປະກາດ array 3 ມິຕິຂຶ້ນມາໃຊງງານດັ່ງຕໍ່ໄປນີ້:

`int num[2][3][4];`

```
example1array2.c example2array2.c example1.c example2.c example5.c example1array3.c
1  #include<stdio.h>
2  #include<stdlib.h>
3  main(){
4      system("color f3");
5      printf("\t===== example 1 array 3 =====\n\n");
6      int num[2][3][4];
7      int i,j,k,count=0;
8      for(i=0;i<2;i++){
9          for(j=0;j<3;j++){
10             for(k=0;k<4;k++){
11                 ++count;
12                 num[i][j][k] = count;
13                 printf("\tnum[%d][%d][%d]=%d\t",i,j,k,num[i][j][k]);
14             }
15             printf("\n");
16         }
17         printf("\n");
18     }
19     printf("\tnum[2][3][4] can keep %d value\n",count);
20     printf("\t===== example 1 array 3 =====\n\n");
21 }
```




ຖາມ ແລະ ຕອບຄໍາຖາມ



