ABES

Estd. 2000

ABES

Engineering College

College Code-032

Data Structure Training

Array – Two Dimensional

Department of CSE, CS & IT

ABES Engineering College, Ghaziabad

Data Structure Training

Array

Address Calculation

Data Structure Training

Array

1 D Array

1000	1002	1004	1006	1008	1010	1012	1014	1016	1018
10	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**	**100**
0	1	2	3	4	5	6	7	8	9

Formula BA+(i-lb)*size of element

int a[10]={10,20,….100};

Base address (BA) = 1000

Find the address of a[5]=?

Loc(a[5])   = 1000 + (5-0)*2

= 1000 + 10

= 1010

Data Structure Training

Array

Example 1.

a[55…..550], BA=990, size of element=10B find the address of a[450].

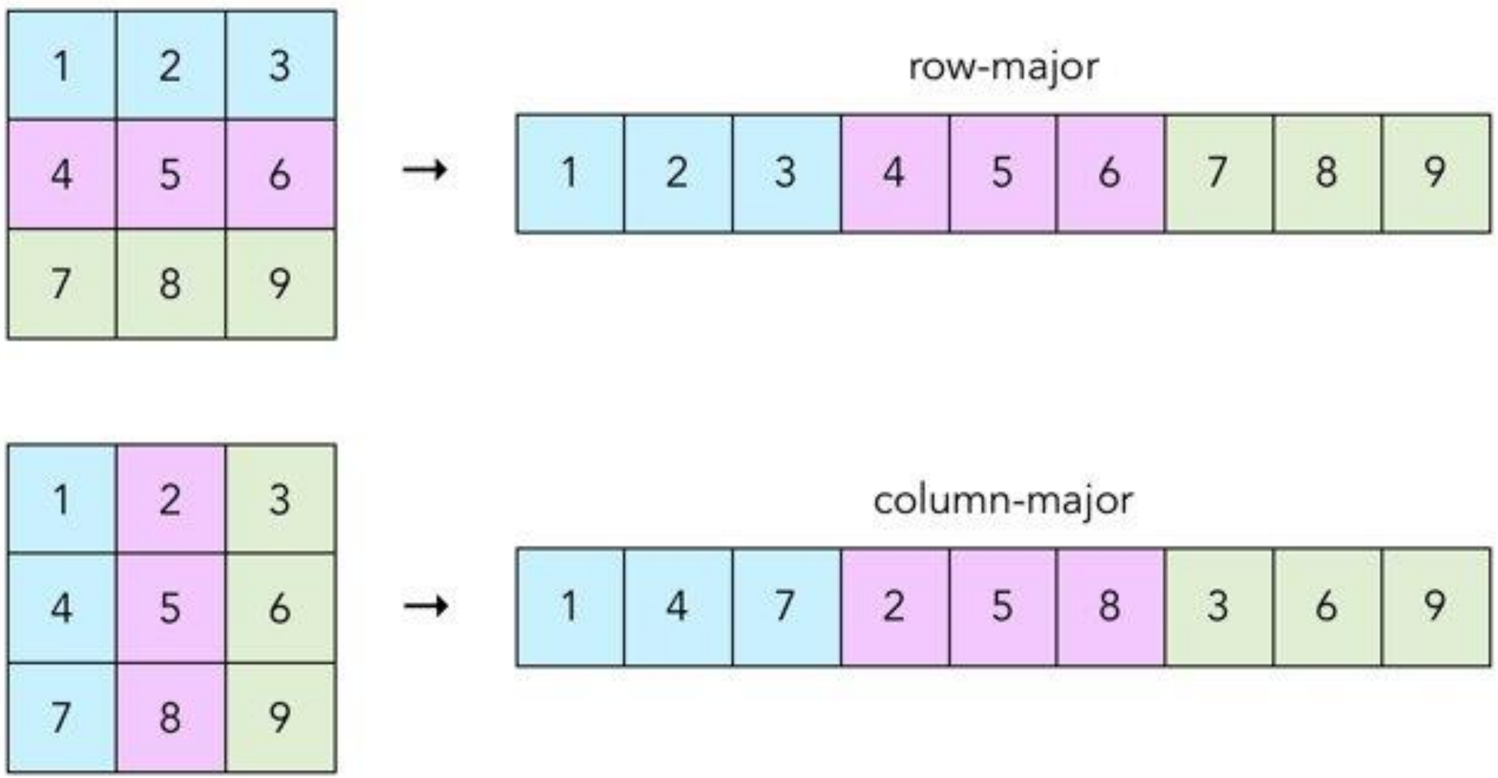
Example 2.

A[-55….55], BA=100, size of element= 5B find the address of a[5].

Data Structure Training

Array

2 D Array



1	2	3
4	5	6
7	8	9

→

row-major

1	2	3	4	5	6	7	8	9

1	2	3
4	5	6
7	8	9

→

column-major

1	4	7	2	5	8	3	6	9

Data Structure Training

Array

Row Major Order

A[lb1……ub1, lb2……ub2]

Formula LOC[i][j] **= BA+ [(i - lb1)* nc + (j - lb2)] * Size**

No. of rows(nr) **=ub1-lb1+1**

No. of columns(nc) **= ub2-lb2+1**

LOC → location/address

BA → Base Address

Size → Size of element

Data Structure Training

Array

Row Major Order

int a[1…4,1….5],

BA = 1000,

find the address of a[4][3]=?

No. of rows(nr) = ub1-lb1+1 = 4-1+1 = 4

No. of columns(nc) = ub2-lb2+1 = 5 – 1 + 1 = 5

Loc(a[4][3]) = 1000 + [(4-1)*5 +(3-1)]*2

= 1000 + [3*5 + 2]*2

= 1034

Data Structure Training

Array

Example 1:

int a[1…4, 1….5], BA = 1000,

find the address of a[2][5]=?

Example 2:

a[25…..750,80….150], BA=1000, C=10,

find the address of a[550][140].

Row Major Order

Example 3:

A[[-25…..25,-50……50], BA= 0, C=1,

find the address of a[20][30].

Data Structure Training

Array

Column Major Order

A[lb1……ub1, lb2……ub2]

Formula LOC[i][j] = BA+ [(j – lb2)* nr + (i – lb1)] * Size

No. of rows(nr) =ub1-lb1+1

No. of columns(nc) = ub2-lb2+1

LOC → location/address

BA → Base Address

Size → Size of element

Data Structure Training

Array

Column Major Order

Int a[1….4,1….5], BA = 1000,

find the address of a[4][3].

No. of rows(nr) **= ub1-lb1+1 = 4-1+1 = 4**

No. of columns(nc) **= ub2-lb2+1 = 5 – 1 + 1 = 5**

LOC(a[4][3])= 1000 + [(3-1)(4-0) + (4-1)]*2

= 1000 + 22

= 1022

Data Structure Training

Array

Column Major Order

Example 1:

Int a[1…4,1….5], BA = 1000,

find the address of a[3][2]=?

Example 2:

a[25…..750,80….150], BA=1000,

C=10, find the address of a[500][90].

Example 3:

A[[-25…..25,-50……50], BA= 0, C=1,

find the address of a[15][20].

Data Structure Training

UGC NET

Array P1

Consider a two dimensional array A[20][10]. Assume 4 words per memory cell, the base address of array A is 100, elements are stored in row-major order and first element is A[0][0]. What is the address of A[11][5] ?

Choose the correct option.

A. 560 **B.** 460

C. 570 **D.** 575

Data Structure Training

UGC NET

Array P1

Consider a two dimensional array A[20][10]. Assume 4 words per memory cell, the base address of array A is 100, elements are stored in row-major order and first element is A[0][0]. What is the address of A[11][5] ?

Choose the correct option.

[✓ **A.** 560]

B. 460

C. 570

D. 575

Data Structure Training

Array P2

Consider a 2D array of integers A[m][n], where m>1 and n>1. If the address of A[1][1] and A[2][1] are 1000 and 1010 respectively and each element occupies 2 bytes then the array has been stored in___________ order.

Choose the correct option.

A. Row major

B. Column major

C. Matrix major

D. Non of these

Data Structure Training

Array P2

Consider a 2D array of integers A[m][n], where m>1 and n>1. If the address of A[1][1] and A[2][1] are 1000 and 1010 respectively and each element occupies 2 bytes then the array has been stored in___________ order.

Choose the correct option.

[✓ **A.** Row major] **B.** Column major

C. Matrix major **D.** Non of these

Data Structure Training

Array

```
#include <stdio.h>
int main(void)
{
int A[][3]={1,2,3,4,5,6,7,8,9};
return 0;
}
```

	0	1	2
0	A[0][0]	A[0][1]	A[0][2]
1	A[1][0]	A[1][1]	A[1][2]
2	A[2][0]	A[2][1]	A[2][2]

A[0][0]	A[0][1]	A[0][2]	A[1][0]	A[1][1]	A[1][2]	A[2][0]	A[2][1]	A[2][2]
1	2	3	4	5	6	7	8	9
1000	1002	1004	1006	1008	1010	1012	1014	1016

Data Structure Training

Array

How to pass a 2D array as a parameter in C?

1. When both dimensions are available globally (as a macro or as a global constant).

```
const int M = 3;
const int N = 3;

void print(int arr[M][N])
{
    int i, j;
    for (i = 0; i < M; i++)
      for (j = 0; j < N; j++)
        printf("%d ", arr[i][j]);
}

int main()
{
    int arr[][3] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};
    print(arr);
    return 0;
}
```

Data Structure Training

Array

How to pass a 2D array as a parameter in C?

2. When only second dimension is available globally (as a macro or as a global constant).

```
const int N = 3;

void print(int arr[][N], int m)
{
        int i, j;
        for (i = 0; i < m; i++)
        for (j = 0; j < N; j++)
                printf("%d ", arr[i][j]);
}

int main()
{
        int arr[][3] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};
        print(arr, 3);
        return 0;
}
```

Data Structure Training

Array

How to pass a 2D array as a parameter in C?

3. Using a single pointer

```
void print(int *arr, int m, int n)
{
        int i, j;
        for (i = 0; i < m; i++)
        for (j = 0; j < n; j++)
                printf("%d ", *((arr+i*n) + j));
}

int main()
{
        int arr[][3] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};
        int m = 3, n = 3;

        print(arr, m, n);
        return 0;
}
```

Data Structure Training

Array P3

```c
#include <stdio.h>
int main(void)
{
int A[][3]={1,2,3,4,5,6,7,8,9};
printf("%u %u",A+1,&A+1);
return 0;
}
```

What would be the output of the above code ? Choose the correct option. Assume array begins at address 200. int data type is of size 4 bytes.

A. 212 236

B. 200 220

C. error

D. None

Data Structure Training

Array P3

```c
#include <stdio.h>
int main(void)
{
int A[][3]={1,2,3,4,5,6,7,8,9};
printf("%u %u",A+1,&A+1);
return 0;
}
```

What would be the output of the above code ? Choose the correct option. Assume array begins at address 200. int data type is of size 4 bytes.

[✓ A. 212 236]

B. 200 220

C. error

D. None

Data Structure Training

Array

```c
#include <stdio.h>
int main(void)
{
int A[][3]={1,2,3,4,5,6,7,8,9};
printf("%u %u %u %u %u %u %u %u", A, *A, A[0], A+0, A[1], A[0]+1, *(A[0]+1), *(*(A+0)+1));
printf("%u %u", *(A+0), *(A+2));
return 0;
}
```

A[0][0]	A[0][1]	A[0][2]	A[1][0]	A[1][1]	A[1][2]	A[2][0]	A[2][1]	A[2][2]
1	2	3	4	5	6	7	8	9
1000	1002	1004	1006	1008	1010	1012	1014	1016

Data Structure Training

Array P4

```
#include <stdio.h>
int main(void)
{
int A[][3]={1,2,3,4,5,6,7,8,9};
printf("%u %u  %u",A[0]+1,*(A[0]+1),*(*(A+0)+1));
return 0;
}
```

What would be the output of the above code ? Choose the correct option. Assume array begins at address 200. int data type is of size 4 bytes.

A. 200 2 2

B. 204 2 2

C. error

D. None

Data Structure Training

Array P4

```c
#include <stdio.h>
int main(void)
{
int A[][3]={1,2,3,4,5,6,7,8,9};
printf("%u %u  %u",A[0]+1,*(A[0]+1),*(*(A+0)+1));
return 0;
}
```

What would be the output of the above code ? Choose the correct option. Assume array begins at address 200. int data type is of size 4 bytes.

A. 200 2 2

[✓ **B.** 204 2 2]

C. error

D. None

Data Structure Training

Array P5

```c
#include <stdio.h>
int main(void)
{
int A[][3]={1,2,3,4,5,6,7,8,9};
printf("%u  %u",*A,*(*A));
return 0;
}
```

What would be the output of the above code ? Choose the correct option. Assume array begins at address 200. int data type is of size 4 bytes.

A. 200 1

B. 1 1

C. error

D. None

Data Structure Training

Array P5

```
#include <stdio.h>
int main(void)
{
int A[][3]={1,2,3,4,5,6,7,8,9};
printf("%u  %u",*A,*(*A));
return 0;
}
```

What would be the output of the above code ? Choose the correct option.
Assume array begins at address 200. int data type is of size 4 bytes.

[✓ A. 200 1]

B. 1 1

C. error

D. None

Data Structure Training

Array P6

```c
#include <stdio.h>
int main()
{
int A[3][3]={1,2,3,4,5,6,7,8,9};
printf("%u %u %d",A,A[2],A[2][2]);
return 0;
}
```

What would be the output of the above code ? Choose the correct option. Assume array begins at address 200. int data type is of size 4 bytes.

A. 200 212 9

B. 200 224 9

C. error

D. None

Data Structure Training

Array P6

```
#include <stdio.h>
int main()
{
int A[3][3]={1,2,3,4,5,6,7,8,9};
printf("%u %u %d",A,A[2],A[2][2]);
return 0;
}
```

What would be the output of the above code ? Choose the correct option. Assume array begins at address 200. int data type is of size 4 bytes.

A. 200 212 9

[✓ **B.** 200 224 9]

C. error

D. None

Data Structure Training

Array P7

```c
#include <stdio.h>
int main()
{
int A[3][3]={1,2,3,4,5,6,7,8,9};
int *p;
p=A;
printf("%u %u %d",A[2],p[2],*(p+2));
return 0;
}
```

What would be the output of the above code ? Choose the correct option. Assume array begins at address 200. int data type is of size 4 bytes.

A. 200 3 3

B. 224 7 7

C. 224 3 3

D. None

Data Structure Training

Array P7

```c
#include <stdio.h>
int main()
{
int A[3][3]={1,2,3,4,5,6,7,8,9};
int *p;
p=A;
printf("%u %u %d",A[2],p[2],*(p+2));
return 0;
}
```

What would be the output of the above code ? Choose the correct option. Assume array begins at address 200. int data type is of size 4 bytes.

A. 200 3 3

B. 224 7 7

[✓ **C.** 224 3 3]

D. None

Data Structure Training

TCS

Array P8

```
B= 100
for i = 1 to n do
    for j = 1 to n do
    {
        Temp = A[i][j] + B
        A[i][j] = A[j][i]
        A[j][i] = Temp - B
    }
for i = 1 to n do
    for j = 1 to n do
        Output(A[i][j]);
```

Let A be a square matrix of size n x n. Consider the following program. What is the expected output?

A. The matrix A itself

B. Transpose of matrix A

C. Adding 100 to the upper diagonal elements and subtracting 100 from diagonal elements of A

D. None of the above

Data Structure Training

TCS

Array P8

```
B= 100
for i = 1 to n do
    for j = 1 to n do
    {
        Temp = A[i][j] + B
        A[i][j] = A[j][i]
        A[j][i] = Temp - B
    }
for i = 1 to n do
    for j = 1 to n do
        Output(A[i][j]);
```

Let A be a square matrix of size n x n. Consider the following program. What is the expected output?

[✓ **A.** The matrix A itself]

B. Transpose of matrix A

C. Adding 100 to the upper diagonal elements and subtracting 100 from diagonal elements of A

D. None of the above

Data Structure Training

ISRO CS

Array P9

An array A consists of n integers in locations A[0], A[1] ....A[n-1]. It is required to shift the elements of the array cyclically to the left by k places, where 1 <= k <= (n-1). An incomplete algorithm for doing this in linear time, without using another array is given below. Complete the algorithm by filling in the blanks. Assume alt the variables are suitably declared

```
min = n; i = 0;
while (____________)
 {
   temp = A[i];
   j = i;
   while (_________)
   {
     A[j] = _________
     j= (j + k) mod n ;
     If ( j< min )
        then min = j;
    }

  A[(n + i — k) mod n] = __________
  i = ___________
```

A. i > min;
   j!= (n+i)mod n;
   A[j + k];
   temp;
   i + 1 ;

B. i < min;
   j!= (n+i)mod n;
   A[j + k];
   temp;
   i + 1;

C. i > min;
   j!= (n+i+k)mod n;
   A[(j + k)];
   temp;
   i + 1;

D. i < min;
   j!= (n+i-k)mod n;
   A[(j + k)mod n];
   temp;
   i + 1;

Data Structure Training

ISRO CS

Array P9

An array A consists of n integers in locations A[0], A[1] ....A[n-1]. It is required to shift the elements of the array cyclically to the left by k places, where 1 <= k <= (n-1). An incomplete algorithm for doing this in linear time, without using another array is given below. Complete the algorithm by filling in the blanks. Assume alt the variables are suitably declared

```
min = n; i = 0;
while (____________)
 {
   temp = A[i];
   j = i;
   while (_________)
   {
     A[j] = _________
     j= (j + k) mod n ;
     If ( j< min )
        then min = j;
    }

 A[(n + i — k) mod n] = __________
 i = ___________
```

A. i > min;
   j!= (n+i)mod n;
   A[j + k];
   temp;
   i + 1 ;

B. i < min;
   j!= (n+i)mod n;
   A[j + k];
   temp;
   i + 1;

C. i > min;
   j!= (n+i+k)mod n;
   A[(j + k)];
   temp;
   i + 1;

D. i < min;
   j!= (n+i-k)mod n;
   A[(j + k)mod n];
   temp;
   i + 1;