Chapter

6A

MEMORY ADDRESS CALCULATION

In this segment, memory address of cells of arrays (two-dimensional) will be calculated. The data-type of the array indicates the size and memory requirement of each cell. The size of each cell may be given as well. Cell addresses are internal values and cannot be input or directly assigned. They can only be calculated by using the reference of any other address. Also to note that the array can store its contents either in Row-major or in Column-major order.

Shown below are few arrays whose cell contents show the memory addresses.

Example 1 : A char array B[7][6] has a base add 2000 at 0,0. The array is stored Column-major wise. Each char requires 2 bytes of storage.

Array diagram showing the memory addresses.

	0	1	2	3	4	5
0	**2000**	2014	2028	2042	2056	2070
1	2002	2016	2030	2044	2058	2072
2	2004	2018	2032	2046	2060	2074
3	2006	2020	2034	2048	2062	2076
4	2008	2022	2036	2050	2064	2078
5	2010	2024	2038	2052	2066	2080
6	2012	2026	2040	2054	2068	2082

Example 2 : An array A[5][6] has a base add 4500 at 0,0. The array is stored Row-major wise. Each cell requires 10 bytes of storage.

Array diagram showing the memory addresses.

	0	1	2	3	4	5
0	**4500**	4510	4520	4530	4540	4550
1	4560	4570	4580	4590	4600	4610
2	4620	4630	4640	4650	4660	4670
3	4680	4690	4700	4710	4720	4730
4	4740	4750	4760	4770	4780	4790

Determining the Address of a Specific Cell in a 2-D Array

The address of a specific cell in a 2-D array can be calculated by using the reference of another cell, storage bytes of each cell and the total number of rows and columns in the array.

There are two ways of calculating cell address :

Method I : By using formula (detail given in the next section)

Method II : By using 2-D array diagram

Method I : Calculating Address of a Specific Cell in a 2D array by using formula

Row major address formula :

$$A[p][q] = B + W \{ tc\ (p - Lr) + (q - Lc) \}$$

Col major address formula :

$$A[p][q] = B + W \{ (p - Lr) + tr \,.\, (q - Lc) \}$$

where—

A[p][q]—Address at row – p and col – q

B – base address

W – storage space in bytes

Lr, Ur – L of r and U of r, Lowest index of Row and Upper index of Row

Lc, Uc – L of c and U of c, Lowest index of Column and Upper index of Column

tc – total num of col → Uc – Lc + 1 (because of the 0 in between)

tr – total num of row → Ur – Lr + 1

Method II : Calculation using Block Diagram

Plot the given values in a 2D array format and fill the cells accordingly. In the following examples, the plotting has been shown.

	0	1	2	3	4	5
0						
1						
2						
3						
4						

In the folloing explain, the plotting has been shown.

Example 1 : A char array B[7][6] has a base add 1046 at 0,0. Cal the add at B[2][3] if the array is stored column major wise. Each char requires 2 bytes of storage.

Col major address formula : $A[p][q] = B + W \{ (p - Lr) + tr \,.\, (q - Lc) \}$

A[p][q]—Address at row - p and col – q

p = 2 **q = 3**

B – base address W – storage space in bytes

B = 1046 **W = 2**

Lr, Ur – L of r and U of r, Lowest index of Row and Upper index of Row

Lc, Uc – L of c and U of c, Lowest index of Column and Upper index of Column

Lr = 0 **Lc = 0** **Ur = 6** **Uc = 5**

tc – total num of col → Uc – Lc +1 → 5 – 0 + 1 = 6

tr – total num of row → Ur – Lr +1 → 6 – 0 + 1 = 7

tr = 7 **tc = 6**

A[p][q] = B + W { (p - Lr) + tr . (q – Lc) }

A[2][3] = 1046 + 2 { (2 - 0) + 7 . (3 – 0) }

= 1046 + 2. (2 + 21)

= 1046 + 2. (23)

= 1046 + 46

= 1092

	0	1	2	3	4	5
0	**1046**	1060	1074	1088		
1	1048	...	...	...		
2	1050	...	...	**1092**		
3	...	...	...			
4	...	...	...			
5	...	...	...			
6	1058	...	...			

Example 2 : Each element of an array A[20][10] requires 2 bytes of storage. If the address of A[6][8] is 4000, find the base address at A[0][0] when the array is stored as **Row Major Wise.**

Row major address formula : A[p][q] = B + W { tc (p – Lr) + (q – Lc) }

A[p][q] - Address at row – p and col – q = 4000

p = 6 **q = 8** B – base address W – storage space in bytes

B = ? W = 2

Lr, Ur – L of r and U of r, Lowest index of Row and Upper index of Row

Lc, Uc – L of c and U of c, Lowest index of Column and Upper index of Column

Lr = 0 **Lc = 0** **Ur = 20** **Uc = 10**

tr – total num of row → Ur – Lr +1 → 19 – 0 + 1 = 20

tc – total num of col → Uc – Lc +1 → 9 – 0 + 1 = 10

tr = 20 **tc = 10**

A[p][q] = B + W { tc (p – Lr) + (q – Lc) }

4000 = B + 2 { 10 (6 – 0) + (8 – 0) }

4000 = B + 2 { 60 + 8 }

4000 = B + 2 { 68 }

4000 = B + 136

B = 4000 – 136

= 3864

	0	1	2	3	4	5	6	7	8	9
0	3864	...	...	...	...	...	...	...	...	...
1	3884	...	...	...	...	...	...	...	...	...
2	3904	...	...	...	...	...	...	...	...	...
3	3924	...	...	...	...	...	...	...	...	...
4	3944	...	...	...	...	...	...	...	...	...
5	3964	...	...	...	...	...	...	...	3980	3982
6	3984	...	...	...	...	...	3996	3998	**4000**	
7										
8										
...										
18										
19										

Example 3 : A 2D array defined as X[3..6, -2..2] requires 2 bytes of storage space for each element. If the array is stored in **Row Major order**, determine the address of X[5, 1], given the base address as 1200.

Row major address formula : X[p][q] = B + W { tc (p – Lr) + (q – Lc) }

X[p][q] – Address at row – p and col – q

p = 5 **q = 1**

B – base address W – storage space in bytes

B = 1200 W = 2

Lr, Ur – L of r and U of r, Lowest index of Row and Upper index of Row

Lc, Uc – L of c and U of c, Lowest index of Column and Upper index of Column

Lr = 3 **Lc = – 2** **Ur = 6** **Uc = 2**

tr – total num of row → Ur – Lr +1 → 6 – 3 + 1 = 4

tc – total num of col → Uc – Lc +1 → 2 – (– 2) + 1 = 5

tr = 4 **tc = 5**

X[p][q] = B + W { tc (p – Lr) + (q – Lc) }

X[5][1] = 1200 + 2 { 5 (5 – 3) + (1 – (– 2)) }

= 1200 + 2 {5 (2) + (1+2) }

= 1200 + 2(10 + 3)

= 1200 + 26

= 1226

	– 2	– 1	0	1	2
3	**1200**	1202	1204	1206	1208
4	...	...	...	...	1218
5	...	...	...	**1226**	
6					

Example 4 : A 2D array defined as AR[– 4..6, – 2..12] stores elements in Row Major Wise, with the address AR[2][3] as 4142. If each element requires 2 bytes of storage, find the Base Address.

Row major address formula : A[p][q] = B + W { tc (p – Lr) + (q – Lc) }

A[p][q]—Address at row - p and col – q B – base address W – storage space in bytes

Lr, Ur – L of r and U of r, Lowest index of Row and Upper index of Row

Lc, Uc – L of c and U of c, Lowest index of Column and Upper index of Column

tc – total num of col → Uc – Lc +1 (because of the 0 in between) → 12 – (-2) + 1 = 15

tr – total num of row → Ur – Lr +1 → 6 – (– 4) + 1 = 11

$$4142 = B + 2 \{ 15 (2 – (– 4)) + (3 – (– 2)) \}$$

$$= B + 2\{15 . 6 + (5) \}$$

$$= B + 2. (95)$$

$$= B + 190$$

$$B = 4142 – 190$$

$$= 3952$$

	– 2	– 1	0	1	2	3	4	5	6	7	8	9	10	11	12
– 4	3952	...	...	...	3960	...	...	...	...	...	...	...	...	...	...
– 3	...	...	...	...	3990	...	...	...	...	...	...	...	...	...	...
– 2	...	...	...	...	4020	...	...	...	...	...	...	...	...	...	...
– 1	...	...	...	...	4050	...	...	...	...	...	...	...	...	...	...
0	...	...	...	...	4080	...	...	...	...	...	...	...	...	...	...
1	...	...	...	...	4110	4112	...	...	4118	4120	4122	4124	4126	4128	4130
2	4132	...	...	...	4140	4142									
3															
4															
5															
6															

Example 5 : A 2D array defined as ZX[4..10, -4..2] requires 5 bytes of storage space for each element. If the array is stored in Row Major order, determine the address of ZX[6, 0], given the base address as 1000.

Row major address formula : X[p][q] = B + W { tc (p – Lr) + (q – Lc) }

X[p][q]—Address at row – p and col – q

p = 6 q = 0

B – base address W – storage space in bytes

B = 1000 W = 5

Lr, Ur – L of r and U of r, Lowest index of Row and Upper index of Row

Lc, Uc – L of c and U of c, Lowest index of Column and Upper index of Column

Lr =4 Lc = -4 Ur = 10 Uc = 2

tr – total num of row → Ur – Lr +1 → 10 – 4 + 1 = 7

tc – total num of col → Uc – Lc +1 → 2 – (-4) + 1 = 7

tr = 7 tc = 7

X[p][q] = B + W { tc (p – Lr) + (q – Lc) }

X[0][0] = 1000 + 5 { 7 (6 – 4) + (0 – (-4)) }

= 1000 + 5 {7 (2) + (4) }

= 1000 + 5 (14 + 4)

= 1000 + 5 (18)

= 1000 + 90

= 1090

	– 4	– 3	– 2	– 1	0	1	2
4	1000	1005	1010	1015	1020	1025	1030
5	1035	...	...	...			
6	...	...	...		1090		
7							
8							
9							
10							

Example 6 : A square matrix A[m x m] is stored in the memory with each element requiring 2 bytes of storage. If the base address A[1][1] is 1098 and the address at A [4][5] is 1144, determine the order of the matrix A[m x m] when the matrix is stored Column Major wise.

Col major address formula :

A[p][q] = B + W { (p – Lr) + tr . (q – Lc) }

1144 = 1098 + 2 { (4 – 1) + tr. (5 -1) } → 46 = 2 { 3 + 4.tr} → 46 = 6 + 8.tr

→ 40 = 8.tr → tr = 5

	1	2	3	4	5
1	1098	1108	1118	1128	1138
2	1100	...	...	...	1140
3	1102	...	...	...	1142
4	1104	...	...	...	1144
5	1106				

Example 7 : A matrix B[10][7] is stored in the memory with each element requiring 2 bytes of storage. If the base address at B [x] [1] is 1012 and the address at B [7][3] is 1060, determine the value 'x' where the matrix is stored in Column Major wise.

Col major address formula :

$$A[p][q] = B + W\{ (p - Lr) + tr . (q - Lc) \}$$

$$1060 = 1012 + 2\{ (7 - x) + 10 (3 - 1) \}$$

$$= 1012 + 2\{ (7 - x) + 20 \}$$

$$= 1012 + 14 - 2x + 40$$

$$= 1066 - 2x$$

$$2x = 1066 - 1060 = 6$$

$$x = 3$$

	1	2	3	4	5	6	7
3	1012	1032	1052	1072			
4	1014	...	1054	...			
5	...	...	1056	...			
6	...	...	1058	...			
7	1020	...	1060	1080			
8	...	...					
9	...	...					
10	...	...					
11	...	...					
12	1030	...					

Example 8 : A 2D array defined as P[4..7, -1..3] requires 5 bytes of storage space for each element. If the array is stored in Column Major order, determine the address of P[6, 2], given the base address as 3500.

Example 9 : A square matrix M [ ][ ] of size 10 is stored in the memory with each element requiring 4 bytes of storage. If the base address at M [0][0] is 1840, determine the address at M [4][8] when the matrix is stored in Row Major Wise.

Example 10 : A square matrix M [ ][ ] of size 10 is stored in the memory with each element requiring 4 bytes of storage. If the base address at M [0][0] is 1840, determine the address at M [4][8] when the matrix is stored in Column Major Wise.

Example 11 : Each element of an array X[-15...10,15...40] requires one byte of storage. If the array is stored in column major order with the beginning location 1500, determine the location of X[5,20].

Example 12 : A two dimensional array defined as x[3...6, -2...2] requires 2 bytes of storage space for each element. If the array is stored in row major order, determine the address of X[5,1], given the base address as 1200.

Example 13 : Each element of an array A[20][10] requires 2 bytes of storage. If the address of A[6][8] is 4000, find the base address at A[0][0] when the array is stored row major wise.

Example 14 : A char array B[7][6] has a base add 1046 at 0,0. If the address at B[4][q] is 1096, find the value of q, given that the array is stored column major wise. Each char requires 2 bytes of storage.

Example 15 : Each element of an array A[5][5] has a base address 1000 at [0][0]. Each cell requires W bytes of storage. If the address of A[3][2] is 1136, find value of W, given the array is stored row major wise.

Table showing memory space requirement by built-in data :

Variable Type	Description	Size
byte	Byte length integer	1 byte
short	Short integer whole numbers	2 bytes
int	Integer whole numbers	4 bytes
long	Bigger integer whole numbers	8 bytes
float	Fractional numbers	4 bytes
double	Bigger fractional numbers	8 bytes
char	Keyboard characters	2 bytes
boolean	Values of type True or False	1 byte

□□