ECONOMICS ASSIGNMENT

CLASS XII C & D

Q.1. What is a supply function?

Q.2. State the law of supply.

Q.3. What is supply schedule?

Q.4. What is market period? What is the shape of supply curve in this period?

Q.5 Why does the supply curve slope upwards?

Q 6. Draw a supply curve of a perishable commodity. Give a reason for the shape of the supply curve.

CLASS XII E

Q.1 Discuss two exceptions to the law of supply.

Q.2 Distinguish between Expansion of supply and Increase in supply.

Q.3 Distinguish between Contraction of supply and Decrease in supply.

Q.4 What will be the effect of increase in input prices on the supply of a commodity X. (Explain with the help of a diagram.)

PHYSICS ASSIGNMENT

Prepare notes from the marked topics in the book.

COMPUTER SCIENCE ASSIGNMENT

ADDRESS CALCULATIONS IN ARRAYS

An Array is a collection of variables of the same type that are referenced by a common name.

Arrays are of different types:-

i. **One Dimensional Arrays**:- comprised of finite homogenous elements.

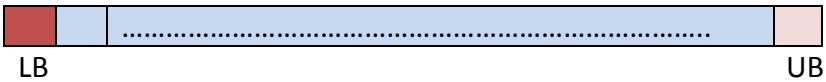
ii. **Multi Dimensional Arrays**:- comprised of elements each of which is itself an array.

SINGLE(ONE) DIMENSIONAL ARRAY

Memory Representation

Internally, arrays are stored as a special object containing:

- A group of contiguous memory location that all have the same name and same data type.
- A reference that stores the beginning address of array elements.
- A separate instance variable containing the number of elements in the array.



- **In Java LB is always 0 and UB = size-1**
- **Array format in Java – Array[size]**

- **In general form LB and UB can be any given +ve/-ve integer.**
- **General Array format: Array[LB..UB]**

Size of Array(Length)= UB – LB + 1

Address of element with index(I)=B+W(I-LB)

Where ***B is Base Address(address of 1st element)***
LB is Lower Bound
UB is Upper Bound
W is element word size (in bytes)

PRACTICE QUESTIONS

Q1. Find the size of an array A[10] declared in Java.

Ans. LB=0
UB=9
Size of array = UB-LB+1
= 9+0+1
=10

Q2. Find the size of an array NUM[-20..15]

Ans. LB = -20
UB= 15
Size of array = UB-LB+1
=15-(-20)+1
=36

Q3. An array ARR[15] is stored in memory with each element requiring 4 bytes of storage. Calculate the address location of ARR[8] when the base address is 100.

Ans. B=100
W= 4bytes
LB= 0
I= 8
Address of ARR[8] = B+W(I-LB)
= 100 + 4(8-0)
= 100 + 32
=132

Q4. An Array A[-10..10] is an integer (long) type. If the beginning location is 150, determine the location of A[5].

Ans. B=150

W=8 bytes (long type)

I=5

LB=-10

UB=10

Address of A[5] = B+W(I-LB)

= 150+8(5-(-10))

=150+8(5+10)

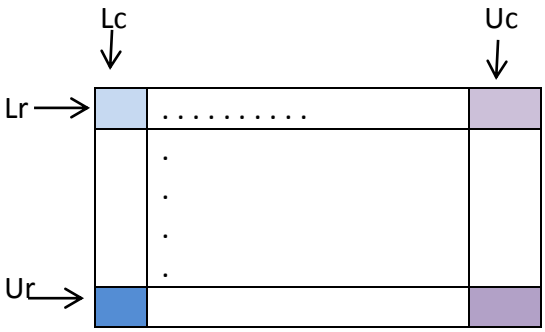
=150+8(15)

=150+20

=270

DOUBLE(MULTI) DIMENSIONAL ARRAY

two-dimensional arrays are stored in row-column matrix, where the first index indicates the row and the second index indicates the column. This means the second index(i.e. column) changes faster than the first index(i.e. row) when accessing the elements in the array in the order in which they are actually stored in memory.



Where

Lr is Lower bound of row

Ur is Upper bound of row

Lc is Lower bound of column

Uc is Lower bound of column

- **In Java Lr and Lc is always 0,**

Ur= No. of rows -1

Uc= No. of columns -1

- **I general Lr, Lc, Ur and Uc can be any +ve/-ve integer**

General Array Format :- Array[Lr..Ur, Lc..Uc]

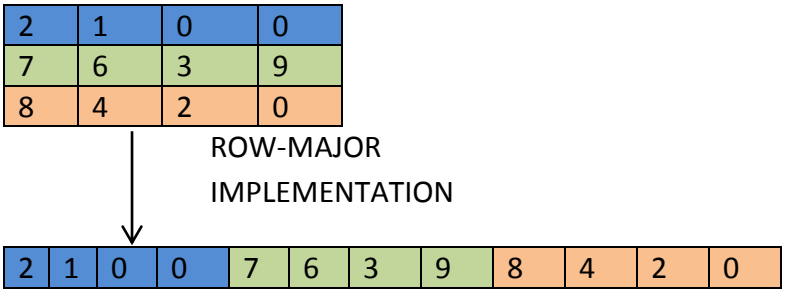
Number of rows (Nr) = Ur-Lr+1
Number of colunms (Nc) = Uc-Lc+1
Number of elements = Nr x Nc

Implementation of Two- Dimensional Array in Memory

While storing the elements of a 2-D arrays in memory, these are allocated contiguous locations. Therefore, a 2-D array must be linearized so as to enable their storage. There are two alternatives to achieve linearization viz. Row-Major and Column –Major.

Row-Major Implementation

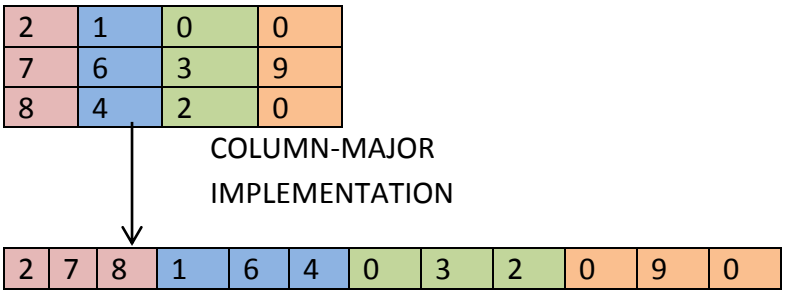
This linearization technique stores values row wise, firstly the first row, then second , then third and so on….



Address of(I,J) th element = B+W [Nc (I – Lr) +(J – Lc)]
Nc = No. of columns = Uc –Lc +1

Column-Major Implementation

This linearization technique stores values column wise, firstly the first column, then second , then third and so on….



Address of(I,J) th element = B+W[Nr (J – Lc) +(J – Lr)]

Nr = No. of rows = Ur –Lr +1

PRACTICE QUESTIONS:

Q1. An array A[5][4] is stored in the memory with each element requiring 4 bytes of storage. If the base address of A[0][0] is 3000, determine the location of A[3][2] when the array is row-major wise.

Ans. Lr=0
Ur= 4 (5-1)
Lc=0
Uc=3 (4-1)
B=3000
I =3
J=2
W= 4 bytes
Nc = Uc-Lc+1=3-0+1=4
Address of A[3][2] element = B+W[Nc(I –Lr)+(J-Lc)]
= 3000 +4 [4(3-0)+(2-0)]
= 3000 +4[12+2]
= 3000+4[14]
=3000+56
=3056

Q2. Each elements of an array A[-5..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 A[5,20].

Ans. Lr=-15
Ur= 10
Lc= 15
Uc= 40
B=1500
I =5
J=20
W= 1 bytes
Nr = Ur-Lr+1= 10-(-15)+1=26
Address of A[5,20] element = B+W[Nr(J–Lc)+(I-Lr)]
= 1500+1[26(20-15)+(5-(-15))]
= 1500+1[130+20]
= 1500+150
=1650

Q3. For a multi-dimensional array B[1..8,-10..5], find the total no of elements in B.

Ans. Lr = 1

Ur = 8

Lc =-10

Uc = 5

Nr = Ur-Lr+1=8-1+1=8

Nc = Uc-Lc+1=5-(-10)+1=16

No. of elements = Nr x Nc

= 8 x16

= 128