Harsh_Kumar_MAN_106_Assignment_1

```
1.  address(A[i])  =  base_address + (i-lb)*(size_of_data_type)

    So
    - A[5]      =    1024 + (5 - (-5))*(size_of_datatype)
    - B[-1][5]  =    512 + (-1 - (-5))*(size_of_datatype)*(10 - (-5) + 1) + (5 - (-5))*(size_of_datatype)

    Assuming size_of_datatype = 4

    - A[5]      = 1064
    - B[-1][5]  = 808

2.
    id   -   4 bytes
    code -   4 bytes
    Name -   8 bytes
    double - 8 bytes

    Size of a node = 24

    - A[1] = 256 + 1*24 = 280
    - A[3].name = 256 + 3*24 + 4 + 4 = 336

    Size of A = 10 * 24 = 240

3.
    &p = 100

    Size of date = 2*4 + 10*1 + 4*4 = 34
    Size of person_detail = 20*1 + 34*2 + 4 + 20*1 + 20*1  = 132

    - p[10]                         = 100 + 10*132 = 1420
    - p[20].address[5]              = 100 + 20*132 + 20 + 2*34 + 4 + 4 = 2836
    - p[20].date_detail[1].month[3] = 100 + 20*132 + 20 + 4*2 + 2*1 = 2770
    - p[20].salary                  = 100 + 20*132 + 20 + 2*34 = 2828
    - p[5].date_detail[2].day[2]    = Index out of range for "day"
    - p[10].emailAddress[10]        = 100 + 10*132 + 20 + 2*34 + 4 + 20 + 9 = 1541

4.
    Notation : A -> Array of size, say N=10
               M -> Number of elements in array
               i -> Index of array where 3 items are to be inserted

    Condition :  i <= m <= N-3

    1. START
    2. Input step      : Take the items to be inserted as a, b, c and take the index in i;
    3. Assignment step : set j = m-1
    4. Loop            : while j >= i ; do 4A and 4B
            4A :         A[j+3] = A[j]
            4B :         j -= 1
    5. Assignment step : set A[i] = a
                         set A[i+1] = b
                         set A[i+2] = c
                         set m = m+3
    7. Stop


5.

    Notation : A -> Array of size, say N=10
               k -> Number of elements in array
               m -> index from which item is to be deleted

    Condition :  m <= k <= N

    1. START
    2. Input step      : Take the index of the value to be deleted at store in m ;
    3. Assignment step : set j = m-1
                         set x = A[m-1]
    4. Loop            : while j<k-1 ; do 4A and 4B
            4A :         A[j] = A[j+1]
            4B :         j += 1
    5. Assignment step : set k = k-1
    6. Return          : return x
    7. Stop
```

```
// 6. A program for reading a two dimensional array row-wise and display its elements column-wise
```

```cpp
#include<bits/stdc++.h>
using namespace std;

#define MAX 3

void coulmn_reading(int *mat, int m, int n){
    for(int i=0; i<m; i++){
        for(int j=0; j<n; j++) cout << mat[i+m*j] << " ";
        cout << '\n';
    }
    cout << '\n';
}

void row_reading(int *mat, int m, int n){
    for(int i=0; i<m; i++){
        for(int j=0; j<n; j++) cout << mat[i*m+j] << " ";
        cout << '\n';
    }
    cout << '\n';
}

int main(){
    int mat[MAX][MAX] ;

    for(int i=0; i<MAX; i++){
        for(int j=0; j<MAX; j++){
            cin >> mat[i][j];
        }
    }

    cout << "Row wise reading\n" ; row_reading(mat[0], MAX, MAX);
    cout << "Coulmn wise reading\n" ; coulmn_reading(mat[0], MAX, MAX);
}
```

```
1 2 3 4 5 6 7 8 9

Row wise reading
1 2 3
4 5 6
7 8 9

Coulmn wise reading
1 4 7
2 5 8
3 6 9
```

```cpp
// 7. Program for reading a two dimensional array and travarsing the array row-wise and display the memory address of array elements.
//    Also Traverse the array column-wise to display the memory address of array elements

#include<bits/stdc++.h>
using namespace std;

#define MAX 3

void coulmn_major(int *mat, int m, int n){
    for(int i=0; i<MAX; i++){
        for(int j=0; j<MAX; j++) cout << &mat[i+m*j] << " ";
    }
    cout << '\n';
}

void row_major(int *mat, int m, int n){
    for(int i=0; i<m*n; i++) cout << &mat[i] << " ";
    cout << '\n';
}

int main(){
    int mat[MAX][MAX] ;

    for(int i=0; i<MAX; i++){
        for(int j=0; j<MAX; j++){
            cin >> mat[i][j];
        }
    }
```

```
    cout << "Row Major\n" ; row_major(mat[0], MAX, MAX);
    cout << "Coulmn Major\n" ; coulmn_major(mat[0], MAX, MAX);
}
```

```
1 2 3 4 5 6 7 8 9

Row Major
0x61fedc 0x61fee0 0x61fee4 0x61fee8 0x61feec 0x61fef0 0x61fef4 0x61fef8 0x61fefc
Coulmn Major
0x61fedc 0x61fee8 0x61fef4 0x61fee0 0x61feec 0x61fef8 0x61fee4 0x61fef0 0x61fefc
```