

Arrays

TOPIC-1

Data Structures and One-dimensional Arrays

Short Answer Type Questions-I[2 Marks Each]

Question 1.

Observe the following C++ code and find out, which out of the given options (i) to (iv) are the expected correct output. Also assign the maximum and minimum value that can be assigned to the variable 'Go'.

```
void main()  
{ int X [4] = {100, 75, 10, 125};  
  int Go = random(2) + 2;  
  for (int i = Go; i<4; i ++)  
    cout<<X [i]<<"$";  
}
```

- (i) 100\$\$75
- (ii) 75\$\$10\$\$125\$\$
- (iii) 75\$\$10\$\$
- (iv) 10\$\$125\$

Answer:

(iv) is correct option
Max. Value = 3
Min. Value = 2

Short Answer Type Questions-II[3 marks each]

Question 1.

Write the definition of a function `grace_score (int score [], int size)` in C ++, which should check all the elements of the array and give an increase of 5 to those scores which are less than 40.

Example: if an array of seven integers is as follows:

45,35,85,80,33,27,90

After executing the function, the array content should be changed as follows :

45,40,85,80,38,32,90

Answer:

```
void grace_score(int score[],int size)
{
    for(int i=0;i<size;i++)
    {
        if (score [i]<40)
            score [i]=score[i]+5;
        cout<< " " ;
    }
}
```

Question 2.

Find the output of the following C++ program:

```
#include<iostream.h> void repch(char s[]) { for(int i=0;s[i]!='\0';i++) { if(((i%2)!=0)
&&(s[i]!=s[i + 1])) { S[i]='@'; cout<<"Hello"; } else if (s[i]==s [i + 1]) { s[i+1]=" !" ; i++ ; } } }
void main() { char str[]="SUCCESS"; cout<<"Original String"<<str; repch(str);
cout<<"Changed String"<<str; }
```

Answer:

Original String SUCCESS
Changed String S@C!ES!

Question 3.

Find the output of the following:

```
#include<iostream.h>
void switchover (int A[],int N, int split)
{
    for(int K=0; K<N; K++)
        if(K<split)
            A [K] + = K;
        else
            A[K] *=K; }
void display(int A [],int N)
{
    for (in K=0; K<N; K++)
        (K%2==0)? Cout<<A[K]<<"%" :
        cout<<A[K]<<endl;
}
void main()
{ int H [ ] = {30, 40, 50, 20, 10, 5};
  switchover(H, 6, 3);
  display(H, 6);
}
```

Answer:

30%41

52% 60

40% 25

Question 4.

An integer array A [30] [40] is stored along the column in the memory. If the element A[20][25] is stored at 50000, find out the location of A[25] [30].

Answer:

$$A[i] [j] = B+W \times [\text{No. of rows} \times (I-L_r) + (J-L_c)]$$

$$A[20] [25] = B + 2 \times [30 \times (20-0) + (25-0)]$$

$$50000 = B + 2 \times [30 \times (20-0) + (25-0)]$$

$$B = 48750$$

$$A[7] [10] = 48750 + 2 \times [30 \times (7-0) + (10-0)]$$

$$= 49190$$

Question 5.

An array P[30][20] is stored along the column in the memory with each element requiring 2 bytes of storage. If the base address of the array P is 26500, find out the location of P[20][10].

Answer:

Total number of rows = 30

Total size = 2 bytes

Base Address = 26500

$$\text{LOC}(P[I][J]) = \text{Base Address} + ((I-LBR) + (J-LBC) \times R) \times W$$

Assuming Lower Bound of Row(LBR)=0

Lower Bound of Column(LBC)=0

Total number of Rows(R)=30

Size of each element(W)=2

$$\text{LOC}(P[20][10]) = 26500 + ((20-0) + (10-0) \times 30) \times 2$$

$$\text{LOC}(P[20][10]) = 26500 + 640$$

$$\text{LOC}(P[20][10]) = 27140$$

Question 6.

Write a function to sort any array of n elements using insertion sort. Array should be passed as argument to the function.

Answer:

```
void insertsort (int a[],int n)
{
```

```

int p,ptr;
//Assuming a[0]=int_min i.e. smallest integer
for(p = 1; p<=n;p++)
{
temp=a[p];
ptr=p-1;
while (temp<a[ptr] )
{
a [ptr+1]=a[ptr] ; //Move Element
Forward
ptr--;
}
a[ptr+1]=temp; //Insert Element in Proper Place
}

```

Question 7.

Write a function NewMAT(int A[],int r, int c) in C++, which accepts a 2d array of integer and its size as parameters divide all those array elements by 6 which are not in the range 60 to 600(both values inclusive) in the 2d Array.

Answer:

```

void NewMAT(int A[] [], int r,int c)
{
for (int i = 0; i<r; i+ + )
for (j=0; j<c;&&(A[i][j]<=600))
A[i][j]/=6

```

OR

```

A[i][j] =A[i][j] /6;
}

```

Question 8.

Write the definition of a function Alter (intA[], int N) in C++, which should change all the multiples of 5 in the array to 5 and rest of the elements as 0. For example, if an array of 10 integers is as follows :

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]
55	43	20	16	39	90	83	40	48	25

After executing the function, the array content should be changed as follows:

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]
5	0	5	0	0	5	0	5	0	5

Answer:

```
void Alter (int A [] , int N)
{
    for (int i=0; i<N; i++)
        if (A[i] %5==0)
            A [i] = 5;
        else
            A [i] = 0;
}
```

OR

Any other correct equivalent function definition

Question 9.

A two dimensional array P[20] [50] is stored in the memory along the row with each of its element occupying 4 bytes, find the address of the element P[10] [30], if the element P[5] [5] is stored at the memory location 15000.

Answer:

Loc (P [I] [J] along the row =BaseAddress+W [(I-LBR)*C+ (J-LBC)]

(where C is the number of columns, LBR = LBC =0)

LOC (P [5] [5]) = BaseAddress + W* [I*C + J]

15000 = BaseAddress + 4 * [5 * 50 + 5]

= BaseAddress + 4 * [250 + 5]

= BaseAddress + 4*255 = BaseAddress + 1020

BaseAddress = 15000 -1020 = 13980

LOC (P [10] [30]) = 13980 + 4* [10 * 50 + 30]

= 13980 + 4 * 530

= 13980 + 2120

= 16100

OR LOC (P [10] [30])

= Loc (P[5] [5]) + W[(I-LBR) * C+ (J-LBC)]

= 15000 + 4 [(10 – 5) * 50 + (30 – 5)]

= 15000 + 4 [5 * 50 + 25]

$$\begin{aligned}
 &= 15000 + 4 * 275 \\
 &= 15000 + 1100 \\
 &= 16100
 \end{aligned}$$

OR

(where C is the number of columns and LBR = LBC = 1)

LOC (P [5] [5])

$$15000 = \text{BaseAddress} + W [(I-1) * C + (J-1)]$$

$$= \text{BaseAddress} + 4 [4*50 + 4]$$

$$= \text{BaseAddress} + 4 [200 + 4]$$

$$= \text{BaseAddress} + 4 * 204$$

$$= \text{BaseAddress} + 816$$

$$\text{BaseAddress} = 15000 - 816 = 14184$$

LOC (P [10] [30])

$$= 14184 + 4 [(10 - 1) * 50 + (30 - 1)]$$

$$= 14184 + 4 [9*50 + 29]$$

$$= 14184 + 4*479$$

$$= 14184 + 1916$$

$$= 16100$$

Question 10.

Write the definition of function Change (int P[], int N) in C+ +, which should change all the multiples of 10 in the array to 10 and rest of the element as 1. For example, if an array of 10 integers is as follows :

P[0]	P[1]	P[2]	P[3]	P[4]	P[5]	P[6]	P[7]	P[8]	P[9]
100	43	20	56	32	91	80	40	45	21

After executing the function, the array content should be changed as follows :

P[0]	P[1]	P[2]	P[3]	P[4]	P[5]	P[6]	P[7]	P[8]	P[9]
10	1	10	1	1	1	10	10	1	1

Answer:

```
void Change (int P[ ] , int N)
{
    for (int i = 0; i<N; i++)
        if (P [i] % 10 == 0)
            P [i] = 10;
        else.
            P [i] = 1;
    }
```

OR

Any other correct equivalent function definition

Question 11.

Write code for a function oddEven (int s[], int N) in C++ , to add 5 in all the odd values and 10 in all the even values of the array S.

Example : If the original content of the array S is

50	11	19	24	28
----	----	----	----	----

The modified content will be :

60	16	24	34	38
----	----	----	----	----

Answer:

```
void oddEven (int s [ ] , int N)
{
    for (int i=0; i<N; i++)
    {
        if(s[i]%2!=0)
            s[i]=s[i]+5;
        else
            s[i]=s [i]+10;
    }
}
```

Question 12.

Write the definition for a function void Transfer (int A[6], int B[6]) in C++, which takes integer arrays, each containing 6 elements as parameters. The function should exchange all odd places (1st, 3rd and 5th) of the two arrays, for example

If the array A contains

15	10	12	21	52	76
----	----	----	----	----	----

And if the array B contains

23	41	67	83	13	53
----	----	----	----	----	----

Then the function should make the contents of the array A as

15	41	12	83	52	53
----	----	----	----	----	----

And the contents of array B as

23	10	67	21	13	76
----	----	----	----	----	----

Answer:

```
void Transfer (int A [6] , int B[6] )  
{  
    int temp;  
    for (int i=;i<6; i+ = 2)  
    {  
        temp=A [i]  
        A [i] =B [i]  
        B [i] +temp;  
    }  
}
```

Question 13.

Write a user-defined function DisTen (int A[] [4], int N, int M) in C++ to find and display all the numbers, which are divisible by 10. For example if the content of array is :

12	20	13
2	10	30

The output should be
20 10 30

Answer:

```
void DisTen (int A[] [4] , int N, int M)
{
    int i, j;
    for (i=0; i<M; i++)
    {
        for (j = 0 ; j <N; j++)
        {
            if (A[i][j]%10==0)
                cout << A [i] [ j] ;
        }
    }
}
```

Question 14.

Write code for a function void ChangeOver (int P[], int N) in C++, which re-positions all the elements of the array by shifting each of them to the next position and by shifting the last element to the first position.

For example: If the content of the array is :

0	1	2	3	4
12	15	17	13	21

The Changed Content will be:

0	1	2	3	4
21	12	15	17	13

Answer:

```
void ChangeOver (int P[] , int N)
{
    int i, j, temp;
    for (i=0; i<N; i = i+1)
    {
        j=j+1;
        temp = P[i];
        P[i] = P [j] ;
        P [j] = temp;
    }
}
```

Question 15.

Write code for a function void Convert (int T[], int Num) in C++, which re-positions all the elements of the array by shifting each of them one to one position before and by shifting the first element to the last position.

For example : If the content of the array is :

0	1	2	3	4
22	25	70	32	12

The Changed Content will be:

0	1	2	3	4
25	70	32	12	22

Answer:

```
void Convert (int T[], int Num)
{
    int i, j, temp;
    for (i=Num; i > 10; i-=1)
    {
        j = i - 1;
        temp = P [ i ] ;
        P [i] = P[j] ;
        P[j] = temp;
    }
}
```

Question 16.

Write a user-defined function DispNTen (int L[] [4], int R, int C) in C++ to find the display all the numbers, which are not divisible by 10. For example if the content of array is :

20	17	30
12	19	10

The output should be
17 12 19

Answer:

```
void DispNTen (int L[] [4], int R,
               int C)
{
```

```

int i, j ;
for (i = 0 ; i<C; i + +)
{for (j = 0; j<R; j+ + )
{
if (A[i] [j]%10==0)
cout <<A[i] [j]<<" ";
}
}
}

```

Question 17.

Write a function Transfer (int A[], int B[], int Size) in C++ to copy the elements of array A into array B in such a way that all the negative elements of A appear in the beginning of B, followed by all the positive elements, followed by all the zeroes maintaining their respective orders in array A. For example :

If the contents of array A are :

7, -23,3,0, -8, -3,4,0

The contents of array B should be

-23, -8, -3,7,3,4,0

Answer:

```

void Transfer (int A[], int B[], int size)
{
for (int i = 0; i<size; i++)
{
if(A[i]<0) B [i] =A [i] ; if (A [i] > 0)
B[i+1]=A[i];
else
B[i/2]=A[i];
}
}

```

Question 18.

A two dimensional array ARR[50] [20] is stored in the memory along the row with each of its elements occupying 4 bytes. Find the address of the element ARR[30][10], if the element ARR[10] [5] is stored at the memory location 15000.

Answer:

Loc (ARR[I] [J] along the row

= BaseAddress + W [(I – LBR) *C + (J – LBC)]

(where C is the number of columns, LBR = LBC = 0 LOC (ARR [10] [5])

= BaseAddress + W [I *C + J]

15000 = BaseAddress + 4 [10 * 20 + 5]

= BaseAddress + 4 [200 + 5]

= BaseAddress + 4 x 205

$= \text{BaseAddress} + 820$
 $\text{BaseAddress} = 15000 - 820$
 $= 14180$
 $\text{LOC}(\text{ARR}[30][10]) = 14180 + 4[30 * 20 + 10]$
 $= 14180 + 4 * 610$
 $= 14180 + 2440$
 $= 16620$
 OR
 $\text{LOC}(\text{ARR}[30][10]) = \text{LOC}(\text{ARR}[10][5]) + W[(I - \text{LBR}) * C + (J - \text{LBC})]$
 $= 15000 + 4[(30-10) * 20 + (10 - 5)]$
 $= 15000 + 4[20 * 20 + 5]$
 $= 15000 + 4 * 405$
 $= 15000 + 1620$
 $= 16620$
 OR
 Where C is the number of columns and $\text{LBR} = \text{LBC} = 1$
 $\text{LOC}(\text{ARR}[10][5])$
 $15000 = \text{BaseAddress} + W[(I - 1) * C + (J - 1)]$
 $= \text{BaseAddress} + 4[9 * 20 + 4]$
 $= \text{BaseAddress} + 4[180 + 4]$
 $= \text{BaseAddress} + 4 * 184$
 $= \text{BaseAddress} + 736$
 $\text{BaseAddress} = 15000 - 736$
 $= 14264$
 $\text{LOC}(\text{ARR}[30][10])$
 $= 14264 + 4[(30 - 1) * 20 + (10 - 1)]$
 $= 14264 + 4[29 * 20 + 9]$
 $= 14264 + 4[580 + 9]$
 $= 14264 + 4 * 589$
 $= 14264 + 2356$
 $= 16620$

Question 19.

Write a function `SORTSCORE()` in C++ to sort an array of structure IPL in descending order of score using selection sort.

Note : Assume the following definition of structure IPL:

struct IPL:

```

{
int Score;
char Teamname[20];
} ;

```

Answer:

```
void SORTSCORE (IPL a[] , int n)
{ int largest;
  IPL temp;
  for ( int K = 0; K <n-1; K++)
  { largest = K;
    for ( int j = K+1; j< n; j++) { if ( a[j].score > a[largest]. score)
    { largest = j;
    }
  }
  temp = a[K];
  a[K] = a[largest];
  a [largest] = temp;
}
```

Question 20.

Write a function in C++ TWOTOONE() which accepts two array X[], Y[] and their size n as argument. Both the arrays X[] and Y[] have the same number of elements. Transfer the content from two arrays X[], Y[] to array Z[]. The even places (0,2,4...) of array Z[] should get the contents from the array X[] and odd places (1,3,5...) of array Z[] should get the contents from the array Y[].

Example : If the X[] array contains 30,60,90 and the Y[] array contains 10,20,50. Then Z[] should contain 30,10,60,20,90,50.

Answer:

```
void TWOTOONE (int x[] , int n)
{
  int z[n] ;
  for (int I =0;i<n;i++)
  {
    if(i%2 == 0)
      z [i] = x [i] ;
    else
      z [i] = y [i] ;
  }
}
```

Question 21.

Write code for a function void EvenOdd (int T[], int C) in C++ , to add 1 in all the odd values and 2 in all the even values of the array T.

Example : If the original content of the array T is

T[0]	T[1]	T[2]	T[3]	T[4]
35	12	16	69	26

The modified content will be:

T[0]	T[1]	T[2]	T[3]	T[4]
36	14	18	70	28

Answer:

```
void EvenOdd (int T [], int C)
{
    for (int i=0; i<C; i++)
    {
        if (T[i] %2==0)
            T[i] += 2;
        else
            T[i] += 1;
    }
}
```

Question 22.

Write a function SWAP2 CHANGE (int P[], int N) in C++ to modify the content of the array in such a way that the elements, which are multiples of 10 swap with the value present in the very next position in the array.

For example:

If the content of array P is

91,50,54,22,30,54

The content of array P should become

91,54,50,22,54,30

Answer:

```
void SWAP 2CHANGE (int P[], int N)
{
    for (int i=0; i<N; i++)
    {
        if (P[i] % 10==0)
        {
            int temp=0;
            temp=P[i];
            P[i]=P[i+1];
            P[i+1]=temp;
        }
    }
}
```

```

P[i+1]=temp;
i++; //Go to next element
}
}
}

```

Question 23.

Write a function SWAP2BEST (int ARR[], int Size) in C++ to modify the content of the array in such a way that the elements, which are multiples of 10 swap with the value present in the very next position in the array.

For example:

If the content of array ARR is

90,56,45,20,34,54

The content of array ARR should become

56,90,45,34,20,54

Answer:

```

void SWAP2BEST (int ARR [] , int Size)
{
for (int n=0; n<size; n++)
{
if (ARR[n]%10 == 0)
{
int temp = 0;
temp = ARR [n];
ARR[n] = ARR[n+1];
ARR [n+1] = temp;
n++;
}
}
}

```

Question 24.

Write the definition of a function AddTax(float Amt[], int N) in C++ , which should modify each element of the array Amt having N elements, as per the following rules :

Existing Value of Amt	Amt to be changed to
If less than 50,000	Add 25% in the existing value
If >=50,000 and <1,00,000	Add 20% in the existing value
If >=1,00,000	Add 15% in the existing value

Answer:

```
void AddTax (Float Amt[] , int N)
{
    for (int i=0; i<N; i++)
        if(Amt[i]<50,000) Amt[i]+=0.25*Amt[i]; else if
        (Amt[i]>=50000&&Amt[i] <100000) Amt[i]+=0.2*Amt[i]; else
        if(Amt[i]>=1,00000)
            Amt[i]+=0.15*Amt[i];
}
```

TOPIC-2

Two-dimensional Arrays

Short Answer Type Questions-I[2 marks each]

Question 1.

Write a function Alternate (int A[] [3], int N, int M) in C + + to display all alternate elements from two-dimensional array A(starting from A[0][0]).

For example:

If the array is containing :

23 54 76

37 19 28

62 13 19

The output will be

23 76 19 62 19

Answer:

```
void Alternative (int A[] [3], intN, int M)
{
    int T=0 ; I<N; I++)
    for (int J=0 ; J {
        if (T%2 = 0)
            COUt << A[I] [J]<<" " ;
        T++ ;
    }
}
```

OR

```
void Alternative (int A[] [3] , intN, int M)
```

```

{
int *P=&A [0] [0] ;
for (int I = 0; I<N*M ; I+=2)
{
Cout<<*p<<" " ;
{ I+=2 ;
}
}
}

```

OR

Any other equivalent correct answer acceptable

Question 2.

Write a function in C++ to print the sum of all the non-negative elements present on both the diagonal of a two dimensional array passed as the argument to the function.

Answer:

```

void Diagonal (int A[] [20] , int N)
{
int K, Sdiag=0;
for(int K=0;K<N;K++) if ( A[K] [K] > 0)
Sdiag+=A [K] [K] ;
for(int K= 0;K<N;K++) if ( A[N-K-1][K] > 0 )
Sdiag+=A[N-K-1] [K] ;
cout<<"Sum of all the non-negative elements on both the
diagonals= "<<Sdiag;
}

```

Question 3.

Write a user-defined function swap_row(int ARR[][3],int R,int C) in C++ to swap the first row values with the last row values : For example:

10	20	30
40	50	60
70	80	90

Then after function call,the content of the array should be:

70	80	90
40	50	60
10	20	30

Answer:

```

void swap_row (int ARR [] [3] , int R,int() )

```

```

{
for(int i=0 J=0: J<C;J++)
{
int temp=ARR[i][j];
ARR[i][j]=ARR[R-1][j];
ARR[R-1][j]=temp;
}
}

```

Question 4.

Write a user-defined function SumLast 3 (int A [] [4], int N, int M) in C + + to find and display the sum of all the values, which are ending with 3 (i.e., units place is 3). Fdr example if the content of array is :

33	13	92
99	3	12

The output should be 49

Answer:

```

void SumLast 3 (int A[] [4], int N, int M)
{
int Sum=0;
for(int i=0; i<N; i++)
for(int j = 0; j <M; j ++ )
{
If ((A[i] [j]-3)% 10==0))
Sum=Sum+A[i][j];
}
cout<<sum;
}

```

Question 5.

Write a user-defined function AddEnd2(int A[] [4], int N, int M) in C+ + to find and display the sum of all the values, which are ending with 2 (i.e., units place is 2). For example, if the content of array is :

22	16	12
19	5	2

The output should be 36

Answer:

```
void AddEnd2(int A[] [4], int N, int M)
{
    int Sum=0;
    for(int i=0; i<N; i++)
        for(int j = 0; j<m;j++)
            if(((A[i] [j]-2)%10 == 0))
                Sum=Sum + A[i] [j] ;
    }
    cout<<sum;
}
```

Question 6.

Write a function in C++ which accepts a 2D array of integers and its size arguments and displays the elements which lie on minor diagonal. [Top right to bottom left diagonal]

[Assuming the 2D array to be square matrix with odd dimension

i.e. 3.3 , 5.5 , 7.7, etc ...]

For example:

If the 2D array is

6 7 8

1 3 6

7 9 3

The following should be displayed :

8

3

7

Answer:

```
void diag (int x[] [] , int m, int n)
{
    for (int i=m; i>0 ; i-- )
    {
        for(int j =0; j<n ; j++)
        {
            if(i==j)
                cout<<" "<<n[i] [j] ;
        }
    }
}
```

}

Question 7.

Write a user-defined function `int SumSingle(int A[4][4])` in C++, which finds and returns the sum of all numbers present in the first row of the array, for example, if the array contains

1	13	10	9
29	17	2	21
14	3	12	31
15	16	25	52

Then the function should return 33.

Answer:

```
int SumSingle (int A[4] [4])
{
    int Sum=0;
    for(int i=0 ; i for (int j=0 ; j Sum=Sum+ A[i] [j]
    return Sum;
}
```

Question 8.

Write a function `SKIPEACH(int H[] [3], int C, int R)` in C++ to display all alternate elements from two-dimensional array H (starting from H [0] [0]).

For example:

If the array is containing:

12 45 67

33 90 76

21 43 59

The output will be

12 67 90 21 59

Answer:

```
void SKIPEACH (int H[ ] [3], int C, int R)
{
    for (int i = 0; i C; i + + )
    { for (int j = 0, j < R; j++)
    {
```

```

if ( (i + j)%2 == 0)
cout<<H[i] [j]<< " " ;
}
}

```

Short Answer Type Questions-II[3 marks each]

Question 1.

An array P[15][10] is stored along the column in the memory with each element requiring 4 bytes of storage. If the base address of array P is 1400, find out the location of P[8][5].

Answer:

$$\begin{aligned} \text{Loc (P[8][5])} &= B + W((I - L_r) + (J - L_c)M) \\ &= 1400 + 4((8 - 0) + (5 - 0)15) \\ &= 1732. \end{aligned}$$

Question 2.

Given an array A[10][12] whose base address is 10000. Calculate the memory location of A[2][5] if each element occupies 4 bytes and array is stored columnwise.

Answer:

$$\begin{aligned} w &= 4 \\ b &= 10000 \\ m &= 10 \\ n[i][j] &= [b + w(i - l_1) + m(j - l_2)] \\ n[2][5] &= [10000 + 4(2 - 0) + 10(5 - 0)] \\ n[2][5] &= [10000 + 8 + 50] \\ n[2][5] &= 10058 \end{aligned}$$

Question 3.

An array T[-1..35][-2..15] is stored in the memory along the row with each element occupying 4 bytes. Find out the base address and address of element T[20][5], if an element T[2][2] is stored at the memory location 3000. Find the total number of elements stored in T and number of bytes allocated to T.

Answer:

Take T[2][2] as the base address [BA],
 sr=starting row, sc=starting column;
 Along the row
 $A[i][j] = BA + \text{size}[nc * (i - sr) + (j - sc)]$
 $T[20][5] = 3000 + 4 [18 * (20 - 2) + (5 - 2)]$

$$\begin{aligned}
&= 3000 + 4[(18 * 18) + (3)] \\
&= 3000 + 4[324 + 3] \\
&= 3000 + 4(327) \\
&= 3000 + 1308 \\
&= 4308 \\
&\text{nc} = \text{no. of columns sr}=2 \text{ sc}=2 \\
&\text{Total number of elements stored in T}=37*18=666 \\
&\text{Total number of bytes allocated to T}=666*4=2664
\end{aligned}$$

Question 4.

An array A[20] [30] is stored along the row in the memory with each element requiring 4 bytes of storage. If the base address of array A is 32000, find out the location of A[15][10]. Also, find the total number of elements present in this array.

Answer:

Array A[20] [30]
 Base Address = 32000
 Size of element(W)=4 bytes
 $A[I][J]=A[15][10]$ hence I=15 and J=10
 Total Row (R)=20
 Total Column (C)=30
 $A[I][J]=B + W[C(I-0)+(J-0)]$
 $A[15][10]=32000 + 4[30(15-0)+(10-0)]$
 $= 32000 + 4[30(15)+10]$
 $= 32000 + 4[450+10]$
 $= 32000 + 4[460]$
 $= 32000 + 1840$
 $= 33840$

Question 5.

An array T[25][20] is stored along the row in the memory with each element requiring 2 bytes of storage. If the base address of array T is 42000, find out the location of T[10][15]. Also, find the total number of elements present in this array.

Answer:

Array A[25] [20]
 Base Address = 42000
 Size of element(W)= 2 bytes
 $A[I][J]=A[10][15]$ hence I=10 and J=15
 Total Row (R) = 25
 Total Column (C)= 20
 $A[I][J]=B + W[C(I-0)+(J-0)]$
 $A[10][15]=42000 + 2[20(10-0)+(15-0)]$
 $= 42000 + 2(20(10+15))$
 $= 42000 + 2[200+15]$

$$\begin{aligned}
&= 42000 + 2(215) \\
&= 42000 + 430 \\
&= 42430
\end{aligned}$$

Question 6.

An array $S[10][15]$ is stored in the memory with each elements requiring 2 bytes of storage. If the base address of array S is 25000, determine the location of $S[5][10]$ if the array is S stored along the column.

Answer:

$S[10][15]$ //Base address; 25000
 $W=2$ bytes
 $S[5][10] = ?$
 $m(\text{row})=0 \ i=5$
 $n(\text{column})=15 \ j=10$
 Array is stored in the memory along the column:
 $P[i][j] = \text{Base address} + u[i+mxj]$
 $s[5][10] = 25000 + 2[5+10 \times 10]$
 $= 25000 + 2(5+100) [1]$
 $= 25000 + 2[105]$
 $= 25000 + 210$
 $= 25210$

Question 7.

An array $T[15][10]$ is stored along the row in the memory with each element requiring 8 bytes of storage. If the base address of array T is 14000, find out the location of $T[10][7]$.

Answer:

Base Address = 14000 = B
 $T[I][J] = B + W [C[I-I_r] + (J-J_c)]$
 $= 14000 + 8 [10(10-0) + (7-0)]$
 $= 14000 + 8 \times 107$
 $= 14000 + 856$
 $= 14856$

Question 8.

Each element of the array $A[8][6]$ is stored using 4 bytes of memory. If the element $A[2][4]$ is stored at location 936, find the address of $A[5][1]$. Assume that the array is stored in Row Major.

Answer:

$n_r=8 \ n_c=6$ size = 4 bytes
 $A[2][4] = 936 \ A[5][1] = ?$

Formula for row-major

$$\begin{aligned}A[i][j] &= BA + \text{size} * ((i - \text{sr}) * \text{nc} + (j - \text{sc})) \\A[5][1] &= A[2][4] + 4 * ((5 - 2) * 6 + (1 - 4)) \\&= 936 + 4 * ((3) * 6 - 3) \\&= 936 + 4 * 15 \\&= 936 + 60 \\A[5][1] &= 996\end{aligned}$$

Question 9.

Write a function REVCOL (in P[] [5], int N, int int M) in C++ to display the content of a two dimensional array, with each column content in reverse order.

Note : Array may contain any number of rows. For example, if the content of array is as follows :

15	12	56	45	51
13	91	92	87	63
11	23	61	46	81

The function should display output as:

11 23 61 46 81

13 91 92 87 63

15 12 56 45 51

Answer:

```
void REVCOL (int P[ ] [5], int N, int M)
{
    for (int I = N - 1; I>=0; I- -)
    {
        for (int J = 0; J<M; J++)
            Cout<<P[I] [J];
        cout<<endl;
    }
}
```

OR

```
void REVCOL (int P [] [5], int N, int M)
{
    for (int I = 0; I<N/2; I++)
    {
        for (int J = 0; J<M; J++)
        {
            int T = P [I] [J] ;
```

```

P [1] [J] = P [N - I - 1] [J] ;
P[N-I-1] [J] = T;
}
}
for (I = 0; I<N; I++)
{
for (int J=0; J<M; J++)
cout<<P [I] [J] ;

cout<<endl;
}
}

```

Question 10.

Write a function REVROW (int N, int M) in C + + to display the content of a two dimensional array, with each row content in reverse order.

For example, if the content of array is as follows:

15	12	56	45	51
13	91	92	87	63
11	23	61	46	81

The function should display output as follows:

```

51 45 56 12 15
63 87 92 91 13
81 46 61 23 81

```

Answer:

```

void REVROW (int P[] [5] , int N, int M)
{
for (int I =0;I<N;I++) {for (int J=M-1; J>=0; J- -)
cout<<P[I] [J];
cout<<endl;
}
}

```

OR

```

void REVROW (int P [] [5] , int N, int M)
{
for (int I=0; I<N; I++)
for (int J=0; J<M/2; J++)
{
int T = P [I] [J] ;

```

```

P [I] [j] = P[I] [M - J - 1] ;
P [I] [M - J - 1] = T;
}
}
for (I = 0; I<N; I++)
{
for (int J = 0; J<M; J++)
Cout<<P[I] [J] ;
cout<<endl;
}
}

```

Question 11.

S[50][20] is a two dimensional array, which is stored in the memory along the row with each of its element occupying 4 bytes, find the address of the element S[5][15], if the element S[8][10] is stored at the memory location 62,000.

Answer:

LOC (S [I] [J])
= Reference Address+W [(I-Lr) *C+ (J-Lc)]
LOC (S [5] [15])
= LOC(5 [8] [10])+4[(15-10)*50+(5-8)]
= 62000 + 4 [5*50-3]
= 62000 + 988
= 62988

Question 12.

Write definition for a function SHOWMID (int P[][5], int R, int C) in C++ to display the elements of first, third and fifth column from a two dimensional array P having R number of rows and C number of columns.

For example, if the content of array is as follows :

515	102	113	801	145
303	141	129	202	121
285	189	100	760	112

The function should display the following as output:

515 303 285
113 129 100
145 121 112

Answer:

```

void SHOWMID (int P[] [5], int R, int C)
{
    for (int J=0; J<C; J++)
        cout<<P[R/2][J]<<" ";
    cout<<endl;
    for (int I=0; I<R; I++)
        cout<<P[I] [C/2]<<" " ;
}

```

Long Answer Type Questions[4 marks each]

Question 1.

T[20][50] is a two dimensional array, which is stored in the memory along the row with each of its element occupying 4 bytes, find the address of the element T[15][5], if the element T[10][8] is stored at the memory location 52000.

Answer:

T [20] [50]

Along the row

$A[i][j] = BA + \text{size} [(i - sr) * nc + j - sc]$

T [10] [8] = 52000

$T[15][5] = 52000 + 4 * ((15 - 10) * 50 + (5 - 8))$

$= 52000 + 4 * (250 + -3)$

$= 52000 + 4 * 247$

$= 52000 + 988$

$= 52988$

Address = 52988

Question 2.

Write definition for a function SHOWMID (int P[][5], int R, int C) in C++ to display the elements of middle row and middle column from a two dimensional array P having R number of rows and C number of columns.

For example, If the content of array is as follows:

115	112	116	101	125
103	101	121	102	101
185	109	109	160	172

The function should display the following as output:

103 101 121 102 101

116 121 109

Answer:

```
void SHOWMID (int P[] [5] , int R, int C)
{
    for(int J=0;J<C;J++)
        cout<<P[R/2] [J] <<" " ;
    cout<<endl;
    for(int I=0;I<R;I++)
        cout<<P[I] [C/2]<<" ";
}
```

Question 3.

R[20][50] is a two dimensional array, which is stored in the memory along the row with each of its element occupying 8 bytes, find the address of the element R[5][15], if the element R[8][10] is stored at the memory location 45,000.

Answer:

R[20] [50]

Along the row

$A[i][j] = BA + \text{size} [(i - Sr)nc + (j - Sc)]$

R[8][10] = 45000

$R[5][15] = 45000 + 8 * ((5 - 8) * 50 + (15 - 10))$

$= 45000 + 8 * (- 3 * 50 + 5)$

$= 45000 + 8 * (-145)$

$= 45000 - 1160$

R[5][15] = 43840