**Chapter Name: Data Structure Arrays (Questions with Answer)**

**Low level**

1. Define a function shift() to shift all odd elements towards left and even to right without changing order of numbers.   
   Example: If input: 1,5,7,8,9,2,10  
   Output: 1,5,7,9,8,2,10.

Ans:

void shift(int a[10],int n)  
{  
int b[10],i,k=0;  
for(i=0;i<n;i++)  
{  
if(a[i]%2= =1)  
{  
b[k]=a[i];  
k++;  
}  
}  
for(i=0;i<n;i++)  
{ if(a[i]%2= =0)  
{ b[k]=a[i];  
k++;  
}  
}  
for(i=0;i<n;i++)  
cout<<b[i]<<" ";  
}

1. Given two arrays A and B. Array ‘A’ contains all the elements of ‘B’ but one more element extra. Write a c++ function which accepts array A and B and its size as arguments/ parameters and find out the extra element in Array A. (Restriction: array elements are not in order)   
   Example : - If Array A is {14, 21, 5, 19, 8, 4, 23, 11} and Array B is {23, 8, 19, 4, 14, 11, 5 } Then output will be 21 (extra element in Array A)

Ans:

void extra(int a[10],int m,int b[10],int n)  
{  
int i,j,flag;  
for(i=0;i<m;i++)  
{ flag=0;  
for(j=0;j<n;j++)  
if(a[i]==b[j])  
{ flag=1;  
break;  
}  
  
if(flag==0)  
cout<<a[i]<<" is the extra element\n";  
}  
}

1. Write C++ function to Arrange(int [],int) to arrange all the negative and positive numbers from left to right.   
   Example : - If an array of 10 elements initially contains { 4,5,6,-7,8,-2,-10,1,13,-20} . Then the function rearrange them in following manner { -20,-10,-7,-2 1,4,5,6,8,13}

Ans:

void arrange(int a[10],int n)  
{  
int i,j,temp;  
for(i=0;i<n;i++)  
{ for(j=0;j<n-1;j++)  
if(a[j]>a[j+1])  
{ temp=a[j];  
a[j]=a[j+1];  
a[j+1]=temp;  
}  
}  
for(i=0;i<n;i++)  
cout<<a[i]<<" ";  
}

1. Write a user defined function in C++ to find and display the row sums of a two dimensional array.

Ans:

void rowsum(int a[10][10],int m,int n)  
{  
int i,j,rsum;  
for(i=0;i<m;i++)  
{ rsum=0;  
for(j=0;j<n;j++)  
rsum+=a[i][j];  
cout<<"\nSum of row "<<i+1<<" is -"<<rsum;  
}  
}

1. Write a user defined function in C++ to find and display the column sums of a two dimensional array.

Ans:

void columnsum(int a[10][10],int m,int n)  
{  
int i,j,csum;  
for(i=0;i<n;i++)  
{ csum=0;  
for(j=0;j<m;j++)  
csum+=a[j][i];  
cout<<"\nSum of column "<<i+1<<" is ->"<<csum;  
}  
}

1. Write a function in C++ to print the product of each row of a two dimensional array passed as the arguments of the function

Ans:

void rowproduct(int a[10][10],int m,int n)  
{  
int i,j,rp;  
for(i=0;i<m;i++)  
{ rp=1;  
for(j=0;j<n;j++)  
rp\*=a[j][i];  
cout<<"\nProduct of row "<<i+1<<" is ->"<<rp;  
}  
}

1. Write a function in C++ which accepts a 2D array of integers and its size as arguments and displays the elements which lie on diagonals. [Assuming the 2D Array to be a square matrix with odd dimension i.e. 3×3, 5×5, 7×7 etc.] Example, if the array content is

5 4 3  
6 7 8  
1 2 9

Output through the function should be: Diagonal 1 : 5 7 9 Diagonal 2 : 3 7 1

Ans:

void diagonal(int a[10][10],int m,int n)  
{  
int i,j;  
cout<<"\nDiagonal 1:";  
for(i=0;i<m;i++)  
{ for(j=0;j<n;j++)  
if(i==j)  
cout<<a[i][j]<<" ";  
}  
cout<<"\n\nDiagonal 2:";  
for(i=0;i<m;i++)  
{ for(j=0;j<n;j++)  
if(i+j==n-1)  
cout<<a[i][j]<<" ";}  
}

1. Write a user defined function in C++ which accepts a squared integer matrix with odd dimensions (3\*3, 5\*5 …) & display the square of the elements which lie on both diagonals. For ex. :

2 5 7   
3 7 2  
5 6 9

The output should be :  
Diagonal one : 4, 49, 81  
Diagonal two : 49, 49, 25

Ans:

void diagonalsquare(int a[10][10],int m,int n)  
{  
int i,j;  
cout<<"\nDiagonal one :";  
for(i=0;i<m;i++)  
{ for(j=0;j<n;j++)  
if(i==j)  
cout<<a[i][j]\*a[i][j]<<",";}  
cout<<"\n\nDiagonal two:";  
for(i=0;i<m;i++)  
{ for(j=0;j<n;j++)  
if(i+j==n-1)  
cout<<a[i][j]\*a[i][j]<<",";  
}  
}

1. Write a user defined function in C++ which accepts a squared integer matrix with odd dimensions (3\*3, 5\*5..)& display the sum of the middle row & middle column elements. For ex. :

2 5 7   
3 7 2  
5 6 9

The output should be :  
Sum of middle row = 12  
Sum of middle column = 18

Ans:

void middlesum(int a[10][10],int m,int n)  
{  
int i,j,mrsum=0,mcsum=0;  
for(i=0;i<n;i++)  
mrsum+=a[m/2][i];  
cout<<"\nSum of middle row="<<mrsum; for(i=0;i<m;i++)  
mcsum+=a[i][n/2];  
cout<<"\nSum of middle column="<<mcsum;  
}

1. Write a user-defined function named Lower\_half() which takes 2D array A, with size N rows and N columns as argument and prints the lower half of the array.   
   Eg. Input

2 3 1 5 0   
7 1 5 3 1   
2 5 7 8 1   
0 1 5 0 1   
3 4 9 1 5

the output will be

2  
7 1  
2 5 7  
0 1 5 0   
3 4 9 1 5

Ans:

void lower\_half(int a[10][10],int n)  
{  
int i,j;  
for(i=0;i<n;i++)  
{ cout<<endl;  
for(j=0;j<n;j++)  
if(i>=j)  
cout<<a[i][j]<<" ";  
}  
}

**MID LEVEL**

1. Write a user-defined function named Upperr\_half() which takes 2D array A, with size N rows and N columns as argument and prints the lower half of the array

Eg.Input

2 3 1 5 0   
7 1 5 3 1   
2 5 7 8 1   
0 1 5 0 1   
3 4 9 1 5

the output will be

2 3 1 5 0  
   1 5 3 1  
     7 8 1   
        0 1   
            5

Ans.

void Upper\_half(int a[10][10],int n)  
{  
int i,j;  
for(i=0;i<n;i++)  
{ cout<<endl;  
for(j=0;j<n;j++)  
if(i<=j)  
cout<<a[i][j]<<" ";  
else  
cout<<” “;  
}  
}

1. Write UDF in C++ which accepts an integer array and its size as arguments/ parameters and assign the elements into a 2 D array of integers in the following format:  
   If the array is 1,2,3,4,5.  
   The resultant 2D array is given below

1 0 0 0 0   
1 2 0 0 0  
1 2 3 0 0  
1 2 3 4 0  
1 2 3 4 5

Ans:

void assign\_lower\_half(int a[10],int n)  
{  
int b[10][10],i,j;  
for(i=0;i<n;i++)  
{  
for(j=0;j<n;j++)  
if(i>=j)  
b[i][j]=a[j];  
else  
b[i][j]=0;  
}  
for(i=0;i<n;i++)  
{ cout<<endl;  
for(j=0;j<n;j++)  
cout<<b[i][j]<<" ";  
}  
}

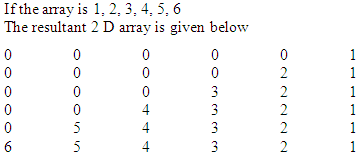
1. Write function in C++ which accepts an integer array and size as arguments and assign values into a 2D array of integers in the following format :  
   If the array is 1, 2, 3, 4, 5, 6  
   The resultant 2D array is given below

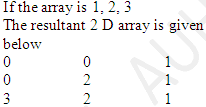
1 2 3 4 5 6  
1 2 3 4 5 0  
1 2 3 4 0 0  
1 2 3 0 0 0  
1 2 0 0 0 0  
1 0 0 0 0 0

Ans:

void r\_upper\_half(int a[10],int n)  
{ int b[10][10],i,j;  
for(i=0;i<n;i++)  
{  
for(j=0;j<n;j++)  
if(i+j<=n-1)  
b[i][j]=a[j];  
else  
b[i][j]=0;  
}  
for(i=0;i<n;i++)  
{ cout<<endl;  
for(j=0;j<n;j++)  
cout<<b[i][j]<<" ";  
}  
}

1. Write a function in C++ which accepts an integer array and its size as arguments/ parameters and then assigns the elements into a two dimensional array of integers in the following format:



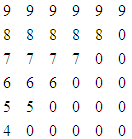


Ans.

void r\_lower\_half(int a[10],int n)  
{  
int b[10][10],i,j,k=n-1;  
for(i=0;i<n;i++)  
{  
for(j=0;j<n;j++)  
if(i+j>=n-1)  
b[j][i]=a[k];  
else  
b[i][j]=0;  
k--;  
}  
for(i=0;i<n;i++)  
{ cout<<endl;  
for(j=0;j<n;j++)  
cout<<b[i][j]<<" ";  
}  
}

1. Write a function in C++ which accepts an integer array and its size as arguments/parameters and assigns the elements into a two dimensional array of integers in the following format.

if the array is 9,8,7,6,5,4 . The resultant 2D array is given below



if the array is 1, 2, 3. The resultant 2D array is given below

http://www.cbseguess.com/papers/cbse_important_questions/xii/2011/images/cs7.gif

Ans:

void r\_upper\_half(int a[10],int n)  
{  
int b[10][10],i,j;   
for(i=0;i<n;i++)  
{  
for(j=0;j<n;j++)  
if(i+j<=n-1)  
b[i][j]=a[i];  
else  
b[i][j]=0;  
}  
for(i=0;i<n;i++)  
{ cout<<endl;  
for(j=0;j<n;j++)  
cout<<b[i][j]<<" ";  
}  
}

1. Write a function in C++ which accepts a integer array and its size as an arguments and prints the output (using nested loops) in following format :  
   Example : if the array is having1 2 4 5 7. Then the output should be

1  
2 2  
4 4 4 4  
5 5 5 5 5  
7 7 7 7 7 7 7

Ans.

void r\_lower\_half(int a[10],int n)  
{  
int i,j;

for(i=0;i<n;i++)  
{ cout<<endl;  
for(j=1;j<=a[i];j++)  
cout<<a[i]<< " ";  
}  
}

1. Write a function in C++ which accepts an integer array and its size as arguments/parameters and assign the elements into a two dimensional array of integers in the following format (size must be odd)

If the array is 1 2 3 4 5 . The output must be

1 0 0 0 5   
0 2 0 4 0   
0 0 3 0 0   
0 2 0 4 0  
1 0 0 0 5

If the array is 10 15 20. The output must be

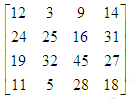
10 0 20  
  0 15 0  
 10 0 20

Ans:

void diagonal(int a[10],int n)  
{  
int b[10][10],i,j,k1=0,k2=n-1;  
for(i=0;i<n;i++)  
for(j=0;j<n;j++)  
{  
if(i==j)  
{  
b[i][j]=a[k1];  
k1++;  
}  
if(i+j==n-1)  
{  
b[i][j]=a[k2];  
k2--;  
}  
if(i!=j&&i+j!=n-1)  
b[i][j]=0;  
}  
for(i=0;i<n;i++)  
{ cout<<endl;  
for(j=0;j<n;j++)  
cout<<b[i][j]<<" "; }  
}

1. Write a function in c++ which accepts a 2D array of integers, number of rows and number of columns as arguments and assign the elements which are divisible by 3 or 5 into a one dimensional array of integers.

If the 2D array is



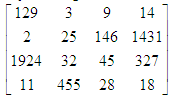
The resultant 1D arrays is 12 , 3 , 9 , 24 , 25 , 45 , 9 , 5 , 18

Ans.

void assign2dto1d(int a[10][10],int m,int n)  
{  
int b[10],i,j,k=0;  
for(i=0;i<m;i++)  
for(j=0;j<n;j++)  
{  
if(a[i][j]%3==0||a[i][j]%5==0)  
{  
b[k]=a[i][j];  
k++;  
}  
}  
cout<<"\nThe resultant 1D array is :";  
for(i=0;i<k;i++)  
{ cout<<b[i]<<" ";}  
}

1. Write a function in C++ which accepts a 2D array of integers and its size as arguments and displays elements which are exactly two digit number.

If 2D array is

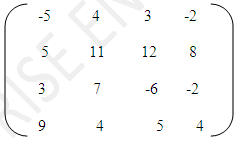


Output is : - 14   25   32   45   11   28   18

Ans.

void twodigit(int a[10][10],int m,int n)  
{  
int i,j; cout<<"\nThe output is: ";  
for(i=0;i<m;i++)  
for(j=0;j<n;j++)  
{  
if(a[i][j]>9 && a[i][j]<100)  
cout<<a[i][j]<<" ";  
}  
}

1. Let A[M X M] be a two dimensional array. Write a function in C++ to find the sum of all the positive elements, which lie on either diagonal. For example, for the matrix shown below, your function should   
   output 43 = (11 + 4 + 12 + 7 + 9):



Ans.

void twodigit(int a[10][10],int m,int n)  
{  
int i,j,sum=0;  
cout<<"\nThe output is: ";  
for(i=0;i<m;i++)  
for(j=0;j<n;j++)  
{ if(i==j||i+j==n-1)  
if(a[i][j]>0)  
sum+=a[i][j];  
}  
cout<<"The output is:"<<sum;  
}

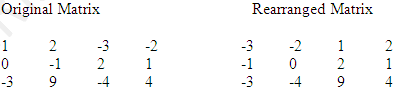
**HIGH LEVEL**

1.Write a function in C++ to replace the repeating elements in an array by 0 . The zeros should be shifted to the end. The order of the array should not change.   
Eg : Array : 10 , 20 , 30 , 10 , 40 , 20 , 30   
Result : 10 , 20 , 30 , 40 , 0 , 0 , 0

Ans.

void twodigit(int a[10],int n)  
{ int i,j,temp;  
for(i=0;i<n;i++)  
for(j=i+1;j<n;j++)  
{ if(a[i]==a[j])  
a[j]=0;  
}  
for(i=0;i<n-1;i++)  
if(a[i]==0)  
{  
temp=a[i];  
a[i]=a[i+1];  
a[i+1]=temp;  
}  
cout<<"\nResult :";  
for(i=0;i<n;i++)  
cout<<a[i]<<" ";  
}

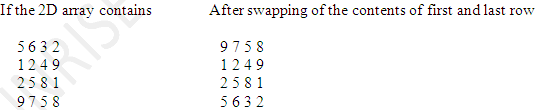
1. Write a function in C++ called shift( ) to rearrange the matrix as shown .



Ans.

void shift(int a[10][10],int m,int n)  
{  
int b[10][10]={0},i,j,k=0,c[10][10]={0};  
for(i=0;i<m;i++)  
{  
k=0;  
for(j=0;j<n;j++)  
{  
if(a[i][j]<0)  
{  
b[i][k]=a[i][j];  
k++;  
}  
}  
} for(i=0;i<m;i++)  
{ k=n-1;  
for(j=n-1;j>=0;j--)  
{  
if(a[i][j]>=0)  
{ c[i][k]=a[i][j];  
k--;  
}  
}  
}  
for(i=0;i<m;i++)  
for(j=0;j<n;j++)  
a[i][j]=b[i][j]+c[i][j];  
for(i=0;i<m;i++)  
{ cout<<endl;  
for(j=0;j<n;j++)  
cout<<a[i][j]<<" ";  
}  
}

1. Define a function SWAPARR( ) in c++ to swap the first row elements with the last row elements, for a 2d integer array passed as the argument of the function.



Ans.

void SWAPARR(int a[10][10],int m,int n)  
{  
int i,j,temp;  
for(i=0;i<n;i++)  
{ temp=a[0][i];  
a[0][i]=a[m-1][i];  
a[m-1][i]=temp;}  
cout<<"\nThe array after swapping";  
for(i=0;i<m;i++)  
{ cout<<endl;  
for(j=0;j<n;j++)  
cout<<a[i][j]<<" ";  
}  
}

1. Write a function in C++ which accepts an integer array M x N and its size as arguments and then displays the array after exchanging the elements in the upper half of the diagonal with the lower half of the diagonal, do not use parallel arrays.

Ans.

void exchange(int a[10][10],int m,int n)  
{ int i,j,temp;  
for(i=0;i<m;i++)  
for(j=0;j<n;j++)  
{ if(i>j)  
{ temp=a[i][j];  
a[i][j]=a[j][i];  
a[j][i]=temp;  
} }  
cout<<"\nThe array after swapping";  
for(i=0;i<m;i++)  
{ cout<<endl;  
for(j=0;j<n;j++)  
cout<<a[i][j]<<" ";  
}}

1. Write a function in C++ which accepts an integer array and its size as arguments and replaces elements having even values with its half and elements having off values with twice its value.  
   Eg. If an array of five elements initially contains the elements as  
   3,4, 5, 16, 9  
   Then the function should rearrange the content of the array as  
   6, 2, 10, 8, 18

Ans.

void change(int a[10],int n)  
{  
int i;  
for(i=0;i<n;i++)  
{  
if(a[i]%2==1)  
a[i]=a[i]\*2;  
else if(a[i]%2==0)  
a[i]=a[i]/2;  
}  
for(i=0;i<n;i++)  
cout<<a[i]<<" ";  
}

1. Define a function COPY( ) to copy the last five elements of array B after first 5 elements of array A and store it in the third array C.

Ans.

void extra(int a[10],int b[10],int n)  
{  
int c[30],counta=0,countb=n-1,countc=0,i;  
while(countc<5)  
{ c[countc]=a[counta];  
counta++;  
countc++;  
}  
while(countc<10)  
{ c[countc]=b[countb];  
countb--;  
countc++;  
}  
for(i=0;i<10;i++)  
cout<<c[i]<<" ";  
}

1. Write a function which finds the locations and values of largest and second largest element in a two dimensional array with M rows and N columns.

Ans.

void largest(int a[10][10],int m,int n)  
{  
int i,j,pr1=0,pc1=0,pr2=0,pc2=0,fl=a[0][0],sl=0;  
for(i=0;i<m;i++)  
for(j=0;j<n;j++)  
{  
if(a[i][j]>fl)  
{  
fl=a[i][j];  
pr1=i;  
pc1=j;  
}  
if((a[i][j]>sl) && a[i][j]<fl)  
{  
sl=a[i][j];  
pr2=i;  
pc2=j;  
}  
}  
cout<<"\nLargest Element is "<<fl<<" and is stored at row "<<pr1<<" and column "<<pc1;  
cout<<"\nSecond Largest Element is "<<sl<<" and is stored at row "<<pr2<<" and column "<<pc2;  
}

8. Write a function in C++ which accepts an integer array of double dimensional with its size as arguments and displays the total numbers of odd, even and prime numbers in the array. Example : if the following integer array will be passed to the function, i.e.

6 4 13 19 5

7 3 8 11 51

9 12 23 4 6

21 29 18 9 10

28 5 12 2 6

Then the output should be : The total odd numbers are : 13

The total odd numbers are : 12

The total odd numbers are : 10

Ans : void numcheck( int arr[ ][ ], int m, int n)

{

int i, j, oddt=0, event=0, primet=0, nf, k;

for(i=0; i<m; i++)

{

for(j=0; j<n; j++)

{

if(arr[i][j] % 2 = = 0)

event++;

else

oddt++;

nf=0;

for(k=1; k<=arr[i][j]; k++)

{

if(arr[i][j] % k = = 0)

nf++;

}

if(nf = = 2)

primet++;

}

}

cout<<”\nThe total odd numbers are : “<<oddt;

cout<<”\nThe total even numbers are : “<<event;

cout<<”\nThe total prime numbers are : “<<primet;

}

1. An array P[20][30] is stored in the memory along the column with each of the element occupying 4 bytes, find out the Base Address of the array, if an element P[2][20] is stored at the memory location 5000.

Ans : Given, W=4, N=20, M=30, Loc(P[2][20])=5000

Column Major Formula:

Loc(P[I][J]) =Base(P)+W\*(N\*J+I)

Loc(P[2][20]) =Base(P)+4\*(20\*20+2)

Base(P) =5000 – 4\*(400+2)

=5000 – 1608

=3392

1. An array ARR[5][25] is stored in the memory with each element occupying 4 bytes of space. Assuming the base address of ARR to be 1000, compute the address of ARR[5][7], when the array is stored as :

(i) Row wise (ii) Column wise.

Ans:

|  |  |
| --- | --- |
| (i) Row wise :  Given, W = 4, N = 5, M = 25,  Base(ARR) = 1000  Row Major Formula:  Loc(ARR[5][7])= Base(ARR)+W\*(M\*I+J)  = 1000 + 4\*( 25\*5 + 7)  = 1000 + 4 \* 132  = 1000 + 528  = 1528 | (ii) Column wise :  Given, W = 4, N = 5, M = 25,  Base(ARR) = 1000  Column Major Formula:  Loc(ARR[5][7])= Base(ARR)+W\*(N\*J+I)  = 1000 + 4\*( 5\*7 + 5)  = 1000 + 4 \* 40  = 1000 + 160  = 1160 |