

## Data Structure (Array, Stack, Queue)

- 1 Write a function in C++ to perform a PUSH operation in a dynamically allocated stack considering the following:
- ```
struct Node
{
    int x,y;
    Node * Link;
};
```
- 2 Write a function in C++ to combine the contents of two equi-sized arrays A and B by computing their corresponding elements with the formula  $2 * A[i] + 3 * B[i]$ , where value I varies from 0 to N-1 and transfer the resultant content in the third same sized array.
- 3 Write a function in C++ to merge the contents of two sorted arrays A and B, into the third array C. Assume array A is sorted in ascending order, B is sorted in descending order, the resultant array is required to be in ascending.
- 4 Write a function in C++ which will accept a 2 D Array of integer and return the sum of all the elements divisible by 5 that lie on the even row number.
- 5 Assume an array A containing elements of structure Accountant is required to be arranged in descending order of salary. Write a C++ program to arrange the same with the help of bubble sort. The array and its size is required to be passed as parameters to the functions. Definition of structure Account is as under:
- ```
struct Account
{
    int Accno;
    char AName[25];
};
```
- 6 Given two arrays of integers x and y of sizes m and n respectively. Write a function named MERGE( ) which will produce a third array named z, such that the following sequence is followed:
- (i) All odds numbers of x from left to right are copied into z from left to right.
  - (ii) All even number of x from left to right are copied into z from right to left.
  - (iii) All odd numbers of y from left to right are copied into z from left to right.
  - (iv) All even number of y from left to right are copied into z from right to left.

Eg: x is {3,2,1,7,6,3}  
 And y is {9,3,5,6,2,8,10}  
 Then z = {3,1,7,3,9,3,5,10,8,2,6,6,2}

- 7 Write a program to multiply two given 2D Matrices using functions.
- 8 Write a user defined function named upperhalf( ) which takes a 2D array A, with size n rows and n cols as arguments and print the upper half of the matrix
- ```
1 2 3           1 2 3
6 7 8           7 8
2 3 4           4
```
- 9 Write a user defined function named lowerhalf( ) which takes a 2D array, with size n, n rows and n cols as arguments and prints the lower half of the matrix.
- Eg;
- ```
1 2 3           1
5 6 7           5 6
9 1 2           9 1 2
```
- 10 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 odd values with twice its value .  
 eg:  
 if the array contains  
 3, 4, 5, 16, 9  
 then the function should be rearranged as  
 6, 2, 10, 8, 18
- 11 Write a function in C++ which accepts an integer array and its size as argument / parameters and assign the elements into a two dimensional array of integers in the following format.  
 If the array is 1 2 3 4 5 6
- ```
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
```
- 12 Consider the following statements
- ```
Char ch,ch1='A';
Char *p,*p1=&ch1;
*p1=ch1+1;
*p=ch;
```
- Suppose each character occupies 2 bytes of memory. If the value assigned to ch is stored in address 0x22 and the value assigned to ch1 is stored in address 0x105 then
- what value is assigned to p1;
  - what is the value of \*p1;
  - what value is assigned to ch?
  - What value is assigned to \*p?
- 13 Write a function in C++ which accepts a 2D array of integers and its size as arguments and displays the elements of middle row and the elements of middle column. Assuming the 2D array to be a square matrix with odd dimensions i.e., 3x3, 5x5, 7x7

eg.,

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

output through the function should be  
middle row 7 8 9  
middle col. 5 8 2

- 14 Define a function `Reversearray(int[], int)` that would accept a one dimensional integer array `NUMBERS` and its size `N`. The function should reverse the content of the array without using any second array.  
Note: use the concept of swapping the elements.  
Eg; if the array initially contains  
2,15,7,8,10,1,13  
after swapping should contain 31,1,10,8,7,15,2
- 15 Given an array named `A` with following elements  
3,-5,1,3,7,0,-15,3,-7,-8  
write a C++ function to shift all the negative numbers to left so that the resultant array may look like  
-5,-15,-7,-8,3,1,3,7,0,3
- 16 Write a function in C++ which accepts an integer array and its size as arguments and swaps the elements of every even location with its odd location  
eg., if the array initially contains  
2, 4, 1, 6, 5, 7, 9, 2, 3, 10  
then it should contain  
4, 2, 6, 1, 7, 5, 2, 9, 10, 3
- 17 From a 2D array `ARR[3][3]` write a program to prepare one dimensions array `ARR2[9]` that will have all the elements of `ARR` as if they are stored in row major form.  
1 2 3  
4 5 6  
7 8 9  
then should contain 123456789
- 18 Consider a 1D array `A` containing `N` integers. Develop an algorithm to do the following.  
(i) Remove all occurrences of a given integer  
(ii) Shift the elements of the array to the right so that unused space is available at the left end.  
(iii) Fill the unused spaces with zero.
- 19 Arrange the following array of integers in ascending order using bubble sort technique.  
Array elements are: 26, 21, 20, 23, 29, 17, 14
- 20 Write a function `check( )` to check if the passed array of 10 integers is sorted or not. The function should return 1 if arranged in ascending order, -1 if arranged in descending order, 0 if it is not sorted.
- 21 Suppose a company keeps single dimensional array `YEAR[100]` such that `YEAR[K]` contains the number of employees appointed in the year `K`. Write a C++ program for each of the following task:  
a. To print each of the years in which no employee was appointed.  
b. To find the number of years in which no employee was appointed.
- 22 Write a user defined function in C++ to find the sum of all positive numbers of a 2D array `ARC`

- [7][7] containing integers.
- 23 Write an interactive menu driven program to create two 3x3 matrices and carry out the following operations.
- Sum of two matrices
  - difference of two matrices
  - Product of two matrices
- Display the input and output matrices in proper matrix form. While creating the matrix keep in mind that each input must not exceed 20.
- 24 Write a function in C++ to perform a PUSH operation in a dynamically allocated stack considering the following;

```
struct Node
{
    int x, y;
    Node *Link;
};
```

- 25 Write a function in C++ to perform insert operation in dynamically allocated Queue containing names of students.
- 26 Write a function in C++ to perform push operation in a dynamically allocated stack containing admission number of students. Also declare the relevant class/ structure and pointers.
- 27 Write a function in C++ to perform a DELETE operation in a dynamically allocated queue considering the following description:

```
Struct Node
{
    float U,V;
    Node *Link;
};
class QUEUE
{
    Node *Rear, *Front;
    Public:
    QUEUE() { Rear =NULL; Front= NULL;}
    void INSERT ();
    void DELETE ();
    ~QUEUE ();
};
```

- 28 Write a function in C++ to perform a PUSH operation in a dynamically allocated stack considering the following :

```
Struct Node
{
    int X,Y;
    Node *Link;
};
class STACK
{
    Node * Top;
    Public:
    STACK() { TOP=NULL;}
    void PUSH();
    void POP();
    ~STACK();
};
```

- 29 Define function stackpush( ) to insert nodes and stackpop( ) to delete nodes, for a linked list implemented stack having the following structure for each node:

```

struct Node
{
    char name[20];
    int age;
    Node *Link;
};
class STACK
{
    Node * Top;
public:
    STACK() { TOP=NULL;}
    void stackpush();
    void stackpop();
    ~STACK();
};

```

- 30 Consider the following portion of a program which implements passengers Queue for a bus. Write the definition of function Insert( ) to insert a new node in the queue with required information.

```

struct Node
{
    float U,V;
    Node *Link;
};
class QUEUE
{
    Node *Rear, *Front;
public:
    QUEUE() { Rear =NULL; Front= NULL;}
    Void INSERT ();
    Void DELETE ();
    ~QUEUE ();
};

```

- 31 Give the necessary declaration of a linked list implemented queue containing float type values . Also write a user defined functions in C++ to add and delete a float type number in the queue.
- 32 An array x[8][20] is stored in the memory with each element requiring 2 bytes of storage. If the base address of the array is 2500, calculate the location of x[5][5] when the array x is stored using the column major order and row major order.
- 33 An array Arr[1..20][1..20] is stored in the memory with each element requiring 4 bytes of storage. If the base address of array Arr is 2000, determine the location of Arr[15][9] when the array Arr is stored in (1) Row wise and (2) Column wise.
- 34 An array MAT[30][10] is stored in the memory row wise with each element occupying 8 bytes of memory. Find out the base address and the address of the element MAT[15][5], if the location of MAT[5][7] is stored at the address 3000.
- 35 An array MAT[20][25] is stored in the memory with each element requiring 2 bytes of storage. If the base address of MAT is 4000 MAT[[12][8] when the array stored in (i) RMO and (ii) CMO
- 36 An array ARR[15][20] is stored in the memory, along the row with each element occupying 4 bytes . Find out the base address and the address of the element ARR[3][2] if the element ARR[5][2] is stored at the address 1500.

- 37 If an array B[11][8] is stored as column wise and B[2][2] is stored at 1024 and B[3][3] at 1084, find the address of B[5][3] and B[1][1].
- 38 An array Arr[50][100] is stored in the memory along the row with each element occupying 2 bytes. Find out the address of the location ARR[20][50] if location of Arr[20][30] is 1350.
- 39 An array x[30][10] is stored in the memory with each element requiring 4 bytes of storage. If the base address of x is 4500, find out memory locations of x[12][8] and x[2][4], if the content is stored along the row.
- 40 An array ARR[15][35] is stored in the memory along the column with each of its elements occupying 8 bytes. Find out the base address and the address of an element ARR[2][5] , if the location is stored at the address 4000
- 41 An array X[15][10] is stored in memory with each element requiring 2 bytes of storage. If the base address of array is 2000, calculate the location of X [7][8] when the array is stored by (1) row major order (2) column major order.
- 42 X [1..6][1...10] is a two dimensional array. The first element of the array is stored at location 100. Each element of the array occupies 6 bytes. Find the memory location of X[2][4] when (i) array is stored row wise. (ii)array is stored column wise
- 43 Each element of an array A[-20..20,10..35] requires one byte of storage. If the array is stored in column major order beginning location 500, determine the location of A[0,30].
- 44 An array S[35][15] is stored in the memory along the row with each of its elements occupying 4 bytes. Find out the memory location for the element S[20][5], if an element S[2][2] is stored at the memory location 3000.
- 45 Given the two dimensional array a[10][20] base address of a is 100 and width of each element is 4 bytes. Find the location of a[8][15] when the array is stored as column-wise and row-wise
- 46 An array A[-2..8][-2..5] is stored in the memory along the column with each element occupying 4 bytes. Find out the address of the element A[3][2].
- 47 An Array Val[1..15][1..10] is stored in the memory with each elements requiring 4 bytes of storage. If the base address of array Val is 1500, determine the location of Val [12][9] when the array Val is stored (i) row wise (ii) column wise.
- 48 A 2-d array defined as A[4..7, -1..3] requires 2 words of storage space for each element. calculate the address of A[6,2], given the base address as 100, also calculate the address of A[7,0] If the array is stored in row major order
- 49 If an array B[11][8] is stored as column wise and B[2][2] is stored at 1024 and B[3][3] at 1084. Find out the base address, size of an element and address of B[5][3].
- 50 An array ARR[35][15] is stored in the memory along the row with each of its element occupying 4 bytes. Find out the base address and the address of an element ARR[20][5], if the location ARR[2][2] is stored at the address 3000.
- 51 An array S[40][30] is stored in the memory along the row with each of the element occupying 4 bytes, find out the memory location for the element S[15][5], if an element s[20][10] is stored at memory location 5700
- 52 An array ARR[10][20] is stored in the memory with each element occupying 2 bytes of space. Assuming the base address of ARR to be 800,compute the address of ARR[9][11], when the array is stored as :
- i) Row wise                          ii) Column wise
- 53 An Array Val[1..15][1..10] is stored in the memory with each elements requiring 4 bytes of storage. If the base address of array Val is 1500, determine the location of Val [12][9] when the array Val is stored (i) row wise (ii) column wise.

- 54 A 2D array defined as  $A[4..7, -1..3]$  requires 2 words of storage space for each element. calculate the address of  $A[6,2]$ , given the base address as 100, also calculate the address of  $A[7,0]$ , if the array is stored in row major order
- 55 An array  $ARR[35][15]$  is stored in the memory along the row with each of its element occupying 4 bytes. Find out the base address and the address of an element  $ARR[20][5]$ , if the location  $ARR[2][2]$  is stored at the address 3000.
- 56 An array  $S[40][30]$  is stored in the memory along the row with each of the element occupying 4 bytes, find out the memory location for the element  $S[15][5]$ , if an element  $s[20][10]$  is stored at memory location 5700.
- 57 An array  $ARR[10][20]$  is stored in the memory with each element occupying 2 bytes of space. Assuming the base address of  $ARR$  to be 800, compute the address of  $ARR[9][11]$ , when the array is stored as :
- i) Row wise
  - ii) Column wise
- 58 An array  $J[15][10]$  is stored in memory with each element requiring two bytes of storage. If the base address of  $J$  is 3000, determine the location of  $J[8][7]$  when the array  $J$  is stored by (i) Row Major, (ii) Column Major.
- 59 An array  $X[1...160][1...10]$  is a two dimensional array. The first element of the array is stored at location 100. Each element of array occupies 6 bytes. Find the memory location of  $X[2][4]$  when array is stored in row wise and column wise.
- 60 Write a function in C++ to find the sum of both left and right diagonal elements from a two dimensional array (Matrix).
- 61 An array  $K$  containing double datatype is arranged in ascending order. Write a user defined function in C++ to search for a number from  $K$  with the help of binary search method. The function should return an integer 0 to show absence of the number and integer 1 to show presence of the number in the array. The function should have the parameters as (i) an array  $K$  (ii) the number  $DATA$  to be searched (iii) number of elements  $N$ .
- 62 Suppose  $X$ ,  $Y$  and  $Z$  are arrays of integers of size  $M$ ,  $N$  and  $M+N$  respectively. The numbers in the array  $X$  appear in ascending order whereas in array  $Y$  are descending order. Write a user defined function to merge the two arrays into third array  $Z$  in ascending order.
- 63 Write a function in C++ which accepts an integer array and its size as arguments and replaces elements having odd values with thrice its value and elements having even values with their squares.
- 64 Write a program to create and display the list of patients in a small hospital using linked list data structure. Each node contains the name of the patient, age and admission number.
- 65 Each node of a stack contains the following information in addition to pointer field:
- (i) Name of the City
  - (ii) Pincode of the City
- Give the Structure of node for the linked stack in question. Top is a pointer that points to the topmost node of the stack. Write the following functions:
- (a)  $PUSH()$ : To push a node into the stack, which is allocated dynamically.
  - (b)  $POP()$ : To remove a node from the stack and release the memory.
- 66 Class Stack
- ```
{
    int data[10];
    int top;
    public:
```

```

        stack ( ) {top = -1;}
        void push( );           // To push an element into the stack
        void pop( );           // To pop an element from the stack
        void Delete(int ITEM ); // To delete all elements that are equal to ITEM
};

```

Complete the class with all function definitions. Use another stack to transfer data temporarily.

- 67 For a circular queue with 5 memory locations, show diagrammatically the status of the queue after each of the following operations:

- (i) 1, 2 and 3 inserted.
- (ii) 1 deleted
- (iii) 4, 5, inserted
- (iv) 2, 3 deleted
- (v) 6 inserted and 4 deleted
- (vi) 7, 8 inserted
- (vii) 5, 6 deleted
- (viii) 11 inserted
- (ix) 7, 8 deleted and 11 deleted

- 68 For a circular queue with 5 memory locations show diagrammatically the status of the FRONT and REAR after each of the following operations:

- (i) E, D and C inserted.
- (ii) E deleted
- (iii) G, H inserted
- (iv) D, C deleted
- (v) F, K inserted
- (vi) G deleted
- (vii) W, X inserted
- (viii) H, F deleted
- (ix) L inserted
- (x) X deleted

- 69 Write an interactive menu driven program to implement circular queue using array consisting of character data value.

- 70 Define member function QueInsert( ) to insert and QueDel( ) to delete nodes of a linked list implemented class Queue having the following Definitions:

```

Struct Node
{
    char name[20];
    int age;
    Node *Link;
};
class Queue
{
    Node *Rear, *Front;
public:
    Queue( ) { Rear=NULL; Front = NULL}
    void QueInsert( );
    void QueDel( );
};

```

- 71 Write a function in C++ to INSERT operation in a dynamically allocated stack considering the following description.

Struct Node



```

{
    float U, V;
    Node *Link;
};

class QUEUE
{
    Node *Rear, *front;
Public:
    QUEUE()
    {
        Rear= NULL;
        Front= NULL;
    }
    void INSERT();
    void DELETE();
};

```

- 72 Write a function that reads 10 integers into the array A. Use another integer array P of same size to store each index of the array A in the following way:  
The index of the first smallest element in A is stored at index 0 of P, the index of the next smallest element in A is stored at index 1 of P and so on. Print the next smallest element in A ordered in the sequence given by each succeeding index stored in P.
- 73 Write a user defined function to sort the given array using Selection sort mechanism.  
int A[] = {10,14,126,23,26,33,44,48,50,55,60,66};  
Print array after each iteration.
- 74 Write a user defined function to search for 55 and 23 in the following array.  
10,14,126,23,26,33,44,48,50,55,60,66  
Make use of binary search method.
- 75 Using a two dimensional A[n x n], write a function to prepare one dimensional array Array[n<sup>2</sup>] that will have all the elements of A as if they were stored in Column major order.
- 76 Write a function to search for a given number in a given array ARR[n] using linear search technique. If the number is found, move it at the top of the array. If the number is not found, insert it at the end of the array.
77. Write a user defined function to sort the array (same as above) using insertion sort in descending order. Give the array status after each iteration
- 78 Write a user defined function to sort the array(same as above) using bubble sort in descending order. Give array status after each iteration
- 79 What would be the output of the following? Assume that the array starts at location 5714 in the memory?  
# include<iostream.h>  
void main()  
{

```

int tab[3][4]={ 5,6,7,8,
               1,2,3,4,
               9,10,0,11};
cout<<"\n"<<*tab[0]<<" "<<*(tab[0]+1);
cout<<"\n"<<*(tab+0)+1);
}

```

80 Write a user defined function to generate the pyramid from string arr("123456789").

1

2 3 2

3 4 5 4 3

4 5 6 7 5 6 4

and so on.