

## ASSIGNMENT 1:

NAME: Vanshaj Goel ROLL NUMBER: 1024030366

QUE1: Develop a Menu driven program to demonstrate the following operations of Arrays

—MENU—

- 1.CREATE
- 2.DISPLAY
- 3.INSERT
- 4.DELETE
- 5.LINEAR SEARCH
- 6.EXIT

CODE:

```
1  #include<bits/stdc++.h>
2  using namespace std;
3
4  void userinput(int arr[], int size){
5      for ( int i =0 ; i<size; i++){
6          cout <<"enter the "<< i+1 << " element "<<endl;           // input from user //
7          cin >> arr[i];
8      }
9  }
10
11 void display(int arr[], int size ){
12     int i =0;
13     while ( i < size ){           // displaying array//
14         cout << arr[i]<<" ";
15         i++;
16     }
17     cout<<endl;
18 }
19
20 void insert(int arr[], int& size , int value , int index){
21     size +=1;           // increasing size of array
22
23     for ( int i = size-1 ; i>= index ; i-- ){           // insertion in array //
24         arr[i+1] = arr[i];
25     }
26     arr[index]= value ;
27 }
28
29 void remove(int arr[], int& size , int index){
30     for ( int i = index ; i< size ; i++ ){           // DELETION in array //
31         arr[i] = arr[i+1];
32     }
33     size -=1;
34 }
35
36 void linearsearch(int arr[], int size , int value) {
37     for (int i =0; i<size; i++){
38         if(arr[i]==value){           // linear search//
39             cout<<" element is present"<<endl;
40             return;
41         }
42     }
43     cout<<"element not present"<<endl;
44 }
```

```

int main(){
    int size = 0;
    int array[100];
    int choice;

    while(choice != 6){
        cout<<"\n---MENU---"<<endl;
        cout<<"1.CREATE"<<endl;
        cout<<"2.DISPLAY"<<endl;
        cout<<"3.INSERT"<<endl;
        cout<<"4.DELETE"<<endl;
        cout<<"5.LINEAR SEARCH"<<endl;
        cout<<"6.EXIT"<<endl;
        cout<<"Enter your choice: ";
        cin >> choice;

        if(choice == 1){
            cout<<"enter the number of elements in array"<<endl;
            cin >> size; // taking input from user//
            userInput ( array , size ) ;
            display ( array , size ) ;
        }
        else if(choice == 2){
            display ( array , size ) ;
        }
        else if(choice == 3){
            cout <<"enter the value to be insterted "<< endl;
            int value ;
            cin >> value ;
            cout <<"enter the index to be insterted "<< endl;
            int indexval;
            cin>> indexval;
            insert ( array , size , value , indexval); // insertion in array
            display ( array , size ) ;
        }
        else if(choice == 4){
            cout <<"enter the index to be deleted "<< endl;
            int indexval;
            cin>> indexval;

            remove ( array , size , indexval); // deletion in array
            display ( array , size ) ;
        }
        else if(choice == 5){

```

```

            cout <<"enter the value to be searched "<< endl;
            int value;
            cin>> value;
            linearsearch(array, size , value); //linear search//
        }
    }
    return 0;
}

```

-----X-----X-----X-----

QUE2: Design the logic to remove the duplicate elements from an Array and after the deletion the array should contain the unique elements.

CODE:

```
1  #include<bits/stdc++.h>
2  using namespace std;
3  void removeduplicate(int arr[], int size , int& res){
4
5      sort(arr,arr + size);
6      int i= 1;
7      while(i<size){
8          if(arr[i]!=arr[res-1]){
9              arr[res]=arr[i];
10             res++;
11         }
12         i++;
13     }
14 }
15 int main(){
16     int array[9]={1,1,2,3,3,4,4,6,6};
17     int res =1;
18     removeduplicate(array , 9, res);
19
20     for( int i =0; i<res; i++){
21         cout <<array[i]<<" ";
22     }
23     return 0;
24 }
```

1 2 3 4 6

-----X-----X-----X-----  
-----  
QUE3: Predict the Output of the following program

```
int main()
{
int i;
int arr[5] = {1};
for (i = 0; i < 5; i++)
printf("%d",arr[i]);
return 0;
}
```

SOL: {1,0,0,0,0}

-----X-----X-----X-----  
-----

QUE4: Implement the logic to  
a. Reverse the elements of an array  
b. Find the matrix multiplication  
c. Find the Transpose of a Matrix

CODE A:

```
1  #include<bits/stdc++.h>
2  using namespace std;
3
4  void display (int arr[] , int size){
5      for (int i = 0; i<size ; i++){
6          cout << arr[i]<<" ";
7      }
8      cout<<endl;
9  }
10
11 void reverse(int arr[], int size){
12     int i =0;
13     int j = size -1;
14     while( i <= j){
15         swap(arr[i], arr[j]);
16         i++;
17         j--;
18     }
19 }
20
21
22
23
24 int main(){
25     cout <<"Array before reversing is "<< endl;
26     int arr[7] = { 1, 2 , 3 , 4, 5 ,6 , 7};
27     display(arr , 7);
28     cout <<"Array after reversing is "<< endl;
29     reverse(arr , 7);
30     display(arr , 7);
31     return 0;
32 }
```

```
Array before reversing is
1 2 3 4 5 6 7
Array after reversing is
7 6 5 4 3 2 1
```

CODE B:

```
1  #include <bits/stdc++.h>
2  using namespace std;
3
4  int main() {
5      int m1[10][10], m2[10][10], m3[10][10];
6      int r1, c1, r2, c2, i, j, k;
7      cout << "Enter rows and cols of first: ";
8      cin >> r1 >> c1;
9      cout << "Enter rows and cols of second: ";
10     cin >> r2 >> c2;
11     if (c1 != r2) {
12         cout << "Not possible!"<<endl;
13         return 0;
14     }
15     cout << "Matrix1:"<<endl;
16     for (i = 0; i < r1; i++) {
17         for (j = 0; j < c1; j++) {
18             cin >> m1[i][j];
19         }
20     }
21     cout << "Matrix2:"<<endl;
22     for (i = 0; i < r2; i++) {
23         for (j = 0; j < c2; j++) {
24             cin >> m2[i][j];
25         }
26     }
27     for (i = 0; i < r1; i++) {
28         for (j = 0; j < c2; j++) {
29             m3[i][j] = 0;
30             for (k = 0; k < c1; k++) {
31                 m3[i][j] += m1[i][k] * m2[k][j];
32             }
33         }
34     }
35     cout << "Result:"<<endl;
36     for (i = 0; i < r1; i++) {
37         for (j = 0; j < c2; j++) {
38             cout << m3[i][j] << " ";
39         }
40         cout << "\n";
41     }
42     return 0;
43 }
```

```
Enter rows and cols of first: 2 2
Enter rows and cols of second: 2 2
Matrix1:
1 2 3 4
Matrix2:
5 6 7 8
Result:
19 22
43 50
```

CODE C:

```
1  #include <bits/stdc++.h>
2  using namespace std;
3
4  int main() {
5      int x[10][10], y[10][10];
6      int r, c, i, j;
7      cout << "Enter rows and cols: ";
8      cin >> r >> c;
9      cout << "Matrix:"<<endl;
10     for (i = 0; i < r; i++) {
11         for (j = 0; j < c; j++) {
12             cin >> x[i][j];
13         }
14     }
15     for (i = 0; i < r; i++) {
16         for (j = 0; j < c; j++) {
17             y[j][i] = x[i][j];
18         }
19     }
20     cout << "Transpose:"<<endl;
21     for (i = 0; i < c; i++) {
22         for (j = 0; j < r; j++) {
23             cout << y[i][j] << " ";
24         }
25         cout << "\n";
26     }
27     return 0;
28 }
```

```
Enter rows and cols: 2 2
Matrix:
1 2 3 4
Transpose:
1 3
2 4
```

-----X-----X-----X-----  
-----  
QUE5: Write a program to find sum of every row and every column in a two-dimensional array.

CODE:

```
1  #include <bits/stdc++.h>
2  using namespace std;
3  int main() {
4      int arr[10][10];
5      int rows, cols;
6      int i, j;
7      int sumrow, sumcol;
8      cout << "Enter rows and columns: ";
9      cin >> rows >> cols;
10     cout << "Enter the elements:"<<endl;
11     for (i = 0; i < rows; i++) {
12         for (j = 0; j < cols; j++) {
13             cin >> arr[i][j];
14         }
15     }
16     for (i = 0; i < rows; i++) {
17         sumrow = 0; // reset for each row
18         for (j = 0; j < cols; j++) {
19             sumrow = sumrow + arr[i][j];
20         }
21         cout << "Sum of row " << i << " = " << sumrow << endl;
22     }
23
24     for (j = 0; j < cols; j++) {
25         sumcol = 0; // reset for each column
26         for (i = 0; i < rows; i++) {
27             sumcol = sumcol + arr[i][j];
28         }
29         cout << "Sum of column " << j << " = " << sumcol << endl;
30     }
31     return 0;}
32
```

```
Enter rows and columns: 2 2
Enter the elements:
1 2 3 4
Sum of row 0 = 3
Sum of row 1 = 7
Sum of column 0 = 4
Sum of column 1 = 6
```

-----X-----X-----X-----  
-----