DATE : __/__/__
PAGE :

{DISHA ; 1024030180}

{Arrays} (C++)

(Assignment-1)

Q1

1) CREATE an Array!

```
#include<iostream>
using namespace std;
int main(){
    int arr[5] = { 1, 2, 3, 4, 5};
    return 0;
}
```

2) Display an Array!

```
#include<iostream>
using namespace std;
int main(){
    int arr[5] = { 1, 2, 3, 4, 5};
    for(int i=0 ; i<n ; i++){
        cout << arr[i] << " ";
    }
    return 0;
}
```

3) Insert an Array!

```
#include<iostream>
using namespace std;
int main(){
    int arr[5] = { 1, 2, 3, 4, 5};
    cout << "original array";
    for(int i=0; i<n; i++){
        cout<< arr[i] << " ";
    }
```

```
int element, position;
cout << "enter element: ";
cin << element;
cout << "enter position where you
        want to insert";
cin << position;

for (int i = n; i > position; i--){
    arr[i] = arr[i-1];
  }
arr[position] = element;
n++;

cout << "New Array: ";
for (int i = 0; i < n; i++){
    cout<<arr[i] << " ";
}
return 0; }
```

4) Delete an Array (elements shifted to left)

```
#include <iostream>
using namespace std;

int main (){
    int arr[5] = { 1, 2, 3, 4, 5};
    int n = 5;
    int position ;
    cout << " Original array : \n";
    for (int i = 0; i < n; i++){
        cout << arr[i] << " ";
    }
cout << " Enter pos" where you want to enter";
cin >> position ;

for (int i = position ; i < n-1; i++){
    arr[i] = arr[i+1];
    }
n--;

cout << " Array after deletion : \n";
for (int i =0; i<n; i++){
    cout << arr[i] << " ";
}
return 0;
}
```

DATE :__/__/__
PAGE :_____

5) Linear Search

```
#include <iostream>
using namespace std;
int search ( int arr[], int sz, int target){
    for( int i= 0; i< sz; i++){
        if(arr[i] == target){
            return i;
        }
    }
    return -1;
}
int main(){
    int arr[] = { 4,2,7,8, 1,2,5};
    int sz = 7;
    int target = 8;
    cout << " search (arr[], sz, target)
    return 0;
}
```

DATE :__/__/__
PAGE :

Q2

```
#include<iostream>
using namespace std;
int removeduplicate(int arr[], int n){
    if(n<=1){
        return n;
    }
    int idx=1;
    for(int i=1; i<n; i++){
        if(arr[i] != arr[i-1]){
            arr[idx++] = arr[i];
        }
    }
    return idx;
}
int main(){
    int arr[]={1,2,2,3,4,4,5};
    int n = sizeof(arr)/sizeof(arr[0]);
    int newsize = removeduplicate(arr,n);
    cout<<"unique size"<<newsize<<"\n";
    cout<<"unique elements:";
    for(int i=0; i<newsize; i++){
        cout<<arr[i]<<" ";
    }
    cout<<"\n";
    return 0;
}
```

DATE :__/__/__
PAGE :_____

Q3 100000

Q4 (a) Reverse an Array:

```
#include <iostream>
using namespace std;

void RevArray(int arr[], int n){
  int t;
  for(int i=0; i<n/2; i++){
      t = arr[i];
      arr[i] = arr[n-i-1];
      arr[n-i-1] = t;
    }
}
int main(){
  int arr[6] = {1, 2, 3, 4, 5, 6};
  int n=6;

  RevArray(arr, n);

  for(int i=0; i<n; i++){
      cout << arr[i] << " ";
  return 0;
}
```

(b)

```
#include<iostream>
using namespace std;
int main(){
    int A[2][2] = {{1,2}, {3,4}};
    int B[2][2] = {{5,6}, {7,8}};
    int result[2][2] = {0};

    for(int i = 0; i < 2 ; i++){
        for(int j = 0; j < 2; j++){
            for(int k=0; k<2 ; k++){
                result[i][j]+= A[i][k]
                               B[k][j]

            }
        }
    }
    cout << " Matrix Multiplication: ";
    for(int i = 0; i < 2; i++){
        for(int j = 0; j < 2; j++){
            cout << result[i][j] << "
        }
        cout << endl;
    }
    return 0;
}
```

DATE :__/__/__
PAGE :_____

(e)

```
#include<iostream>
using namespace std;
int main(){
    int rows = 2, cols = 3;
    int matrix[2][3] = {{1,2,3},{4,5,6}};
    int transpose[3][2];

    for(int i = 0; i < rows; i++){
        for(int j = 0; j < cols; j++){
            transpose[j][i] = matrix[i][j];
        }
    }
    cout << "Transpose of matrix" << endl;
    for(int i = 0; i < cols; i++){
        for(int j = 0; j < rows; j++){
            cout << transpose[i][j] << " ";
        }
        cout << endl;
    }
    return 0;
}
```

DATE :__/__/__
PAGE :_____

Q8

```
#include <iostream>
using namespace std;
int main() {
    int rows = 3, cols = 3;
    int arr[3][3] = {
        {1, 2, 3}, {4, 5, 6}, {7, 8, 9} };

    cout << "sum of each row" ;
    for (int i = 0; i < rows ; i++){
        int rowsum = 0;
        for (int j = 0; j < cols < j++){
            rowsum += arr[i][j];
        }
        cout << "Row" << i+1 << ": "
            << rowsum << \n ;
    }

    cout << " sum of each column :";
    for (int j = 0; j < cols ; j++){
        int colsum = 0;
        for (int i = 0 ; i < rows; i++)
            colsum += arr[i][j];
        }
        cout << "column" << j+1 << " :"
            << colsum << \n;
    }
    return 0;
}
```