ASSIGMENT 1

QUESTION 1

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

#define MAX 100

class ArrayOperations {
    int arr[MAX];
    int size;

public:
    ArrayOperations() {
        size = 0;
    }

    // CREATE
    void create() {
        cout << "Enter number of elements: ";
        cin >> size;
        cout << "Enter elements:\n";
        for (int i = 0; i < size; i++) {
            cin >> arr[i];
        }
    }

    // DISPLAY
    void display() {
```

```
    if (size == 0) {
        cout << "Array is empty!\n";
        return;
    }
    cout << "Array elements are: ";
    for (int i = 0; i < size; i++) {
        cout << arr[i] << " ";
    }
    cout << endl;
}

// INSERT
void insert() {
    int pos, val;
    if (size == MAX) {
        cout << "Array is full!\n";
        return;
    }
    cout << "Enter position (1-" << size+1 << "): ";
    cin >> pos;
    cout << "Enter value to insert: ";
    cin >> val;

    if (pos < 1 || pos > size + 1) {
        cout << "Invalid position!\n";
        return;
    }
    for (int i = size; i >= pos; i--) {
        arr[i] = arr[i - 1];
    }
    arr[pos - 1] = val;
```

```
    size++;
    cout << "Element inserted.\n";
}

// DELETE
void Delete() {
    int pos;
    if (size == 0) {
        cout << "Array is empty!\n";
        return;
    }
    cout << "Enter position (1-" << size << "): ";
    cin >> pos;
    if (pos < 1 || pos > size) {
        cout << "Invalid position!\n";
        return;
    }
    for (int i = pos - 1; i < size - 1; i++) {
        arr[i] = arr[i + 1];
    }
    size--;
    cout << "Element deleted.\n";
}

// LINEAR SEARCH
void linearSearch() {
    int key, flag = 0;
    cout << "Enter element to search: ";
    cin >> key;
    for (int i = 0; i < size; i++) {
        if (arr[i] == key) {
```

```
            cout << "Element found at position " << (i + 1) << endl;
            flag = 1;
            break;
        }
    }
    if (!flag) {
        cout << "Element not found!\n";
    }
  }
};

int main() {
  ArrayOperations obj;
  int choice;

  do {
    cout << "\n--MENU--\n";
    cout << "1.CREATE\n2.DISPLAY\n3.INSERT\n4.DELETE\n5.LINEAR SEARCH\n6.EXIT\n";
    cout << "Enter your choice: ";
    cin >> choice;

    switch (choice) {
    case 1: obj.create(); break;
    case 2: obj.display(); break;
    case 3: obj.insert(); break;
    case 4: obj.Delete(); break;
    case 5: obj.linearSearch(); break;
    case 6: cout << "Exiting...\n"; break;
    default: cout << "Invalid choice!\n";
    }
  } while (choice != 6);
```

```
    return 0;
}
```

QUESTION 2

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

int main() {
   int arr[100], n;
   cout << "Enter number of elements: ";
   cin >> n;

   cout << "Enter elements:\n";
   for (int i = 0; i < n; i++) {
      cin >> arr[i];
   }

   // Removing duplicates
   for (int i = 0; i < n; i++) {
      for (int j = i + 1; j < n; j++) {
         if (arr[i] == arr[j]) {
            // Shift elements left
            for (int k = j; k < n - 1; k++) {
               arr[k] = arr[k + 1];
            }
            n--;    // reduce array size
            j--;    // check again at same index
         }
      }
   }

   cout << "Array after removing duplicates: ";
   for (int i = 0; i < n; i++) {
      cout << arr[i] << " ";
```

```
    }
    cout << endl;

    return 0;
}
```

QUESTION 3

3)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:- 11111

QUESTION 4

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

class Operations {
public:
    // Reverse an array
    void reverseArray() {
        int n, arr[100];
        cout << "Enter size of array: ";
        cin >> n;

        cout << "Enter elements: ";
        for (int i = 0; i < n; i++) cin >> arr[i];

        for (int i = 0, j = n - 1; i < j; i++, j--) {
            int temp = arr[i];
            arr[i] = arr[j];
            arr[j] = temp;
        }

        cout << "Reversed array: ";
        for (int i = 0; i < n; i++) cout << arr[i] << " ";
        cout << endl;
    }

    // Matrix multiplication
    void matrixMultiplication() {
        int m, n, p;
        cout << "Enter rows and cols of first matrix (m n): ";
        cin >> m >> n;
```

```
cout << "Enter cols of second matrix (p): ";
cin >> p;

int A[50][50], B[50][50], C[50][50] = {0};

cout << "Enter elements of first matrix (" << m << "x" << n << "):\n";
for (int i = 0; i < m; i++)
   for (int j = 0; j < n; j++)
      cin >> A[i][j];

cout << "Enter elements of second matrix (" << n << "x" << p << "):\n";
for (int i = 0; i < n; i++)
   for (int j = 0; j < p; j++)
      cin >> B[i][j];

// multiplication
for (int i = 0; i < m; i++) {
   for (int j = 0; j < p; j++) {
      C[i][j] = 0;
      for (int k = 0; k < n; k++) {
         C[i][j] += A[i][k] * B[k][j];
      }
   }
}

cout << "Resultant Matrix (" << m << "x" << p << "):\n";
for (int i = 0; i < m; i++) {
   for (int j = 0; j < p; j++) {
      cout << C[i][j] << " ";
   }
   cout << endl;
```

```
        }
    }

    // Transpose of a matrix
    void transposeMatrix() {
        int m, n, A[50][50], T[50][50];
        cout << "Enter rows and cols of matrix: ";
        cin >> m >> n;

        cout << "Enter elements of matrix (" << m << "x" << n << "):\n";
        for (int i = 0; i < m; i++)
            for (int j = 0; j < n; j++)
                cin >> A[i][j];

        for (int i = 0; i < m; i++) {
            for (int j = 0; j < n; j++) {
                T[j][i] = A[i][j];
            }
        }

        cout << "Transpose Matrix (" << n << "x" << m << "):\n";
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < m; j++) {
                cout << T[i][j] << " ";
            }
            cout << endl;
        }
    }
};

int main() {
```

```
    Operations op;
    int choice;

    do {
        cout << "\n-- MENU --\n";
        cout << "1. Reverse Array\n";
        cout << "2. Matrix Multiplication\n";
        cout << "3. Transpose of Matrix\n";
        cout << "4. Exit\n";
        cout << "Enter your choice: ";
        cin >> choice;

        switch (choice) {
            case 1: op.reverseArray(); break;
            case 2: op.matrixMultiplication(); break;
            case 3: op.transposeMatrix(); break;
            case 4: cout << "Exiting...\n"; break;
            default: cout << "Invalid choice!\n";
        }
    } while (choice != 4);

    return 0;
}
```

QUESTION 5

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

int main() {
    int m, n;
    cout << "Enter number of rows: ";
    cin >> m;
    cout << "Enter number of columns: ";
    cin >> n;

    int A[50][50];

    cout << "Enter elements of the matrix (" << m << "x" << n << "):\n";
    for (int i = 0; i < m; i++) {
        for (int j = 0; j < n; j++) {
            cin >> A[i][j];
        }
    }

    // Sum of each row
    cout << "\nSum of each row:\n";
    for (int i = 0; i < m; i++) {
        int rowSum = 0;
        for (int j = 0; j < n; j++) {
            rowSum += A[i][j];
        }
        cout << "Row " << i + 1 << " = " << rowSum << endl;
    }

    // Sum of each column
```

```
    cout << "\nSum of each column:\n";
    for (int j = 0; j < n; j++) {
        int colSum = 0;
        for (int i = 0; i < m; i++) {
            colSum += A[i][j];
        }
        cout << "Column " << j + 1 << " = " << colSum << endl;
    }

    return 0;
}
```