

Q1

```
#include <iostream>

using namespace std;

#define MAX_SIZE 100

int arr[MAX_SIZE];

int n = 0;

void createArray() {

    cout << "Enter number of elements: ";

    while (!(cin >> n) || n < 0 || n > MAX_SIZE) {

        cin.clear();

        cin.ignore(10000, '\n');

        cout << "Invalid input! Enter a number between 0 and " << MAX_SIZE << ":";

    }

    cout << "Enter " << n << " elements: ";

    for (int i = 0; i < n; i++) {

        while (!(cin >> arr[i])) {

            cin.clear();

            cin.ignore(10000, '\n');

            cout << "Invalid input! Re-enter element " << i + 1 << ":";

        }

    }

}
```

```
}
```

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

```
void insertElement() {
    if (n >= MAX_SIZE) {
        cout << "Array is full! Cannot insert." << endl;
        return;
    }
    int pos, val;
    cout << "Enter position (0 to " << n << "): ";
    while (!(cin >> pos) || pos < 0 || pos > n) {
        cin.clear();
        cin.ignore(10000, '\n');
        cout << "Invalid position! Enter again: ";
    }
    cout << "Enter value: ";
```

```
while (!(cin >> val)) {  
  
    cin.clear();  
  
    cin.ignore(10000, '\n');  
  
    cout << "Invalid input! Enter again: ";  
  
}  
  
for (int i = n; i > pos; i--) arr[i] = arr[i - 1];  
  
arr[pos] = val;  
  
n++;  
  
cout << "Element inserted." << endl;  
  
}
```

```
void deleteElement() {  
  
    if (n == 0) {  
  
        cout << "Array is empty!" << endl;  
  
        return;  
  
    }  
  
    int pos;  
  
    cout << "Enter position (0 to " << n - 1 << "): ";  
  
    while (!(cin >> pos) || pos < 0 || pos >= n) {  
  
        cin.clear();  
  
        cin.ignore(10000, '\n');  
  
        cout << "Invalid position! Enter again: ";  
  
    }  
  
    for (int i = pos; i < n - 1; i++) arr[i] = arr[i + 1];  
  
    n--;
```

```
cout << "Element deleted." << endl;
}

void linearSearch() {
    if (n == 0) {
        cout << "Array is empty!" << endl;
        return;
    }
    int key;
    cout << "Enter element to search: ";
    while (!(cin >> key)) {
        cin.clear();
        cin.ignore(10000, '\n');
        cout << "Invalid input! Enter again: ";
    }
    for (int i = 0; i < n; i++) {
        if (arr[i] == key) {
            cout << "Element found at position " << i << endl;
            return;
        }
    }
    cout << "Element not found." << endl;
}

int main() {
```

```
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: ";

    if (!(cin >> choice)) {

        cin.clear();

        cin.ignore(10000, '\n');

        cout << "Invalid choice! Try again.\n";

        continue;

    }

    switch (choice) {

        case 1: createArray(); break;

        case 2: displayArray(); break;

        case 3: insertElement(); break;

        case 4: deleteElement(); break;

        case 5: linearSearch(); break;

        case 6: cout << "Exiting program.\n"; break;

        default: cout << "Invalid choice! Try again.\n";

    }

} while (choice != 6);

return 0;
}
```

```
---- MENU ----
1. CREATE
2. DISPLAY
3. INSERT
4. DELETE
5. LINEAR SEARCH
6. EXIT
Enter your choice: 1
Enter number of elements: 2
Enter 2 elements: 4 5

---- MENU ----
1. CREATE
2. DISPLAY
3. INSERT
4. DELETE
5. LINEAR SEARCH
6. EXIT
Enter your choice: █
```

Q2

```
#include <iostream>

using namespace std;

int main() {
    int size;
    cout << "Enter the size of the array: ";
    while (!(cin >> size) || size <= 0) {
        cin.clear();
        cin.ignore(10000, '\n');
        cout << "Invalid size! Enter again: ";
    }
}
```

```
int arr[size];

cout << "Enter the elements of the array: ";

for (int i = 0; i < size; i++) {

    while (!(cin >> arr[i])) {

        cin.clear();

        cin.ignore(10000, '\n');

        cout << "Invalid input! Enter again: ";

    }

}

cout << "The unique array is: ";

for (int i = 0; i < size; i++) {

    bool duplicate = false;

    for (int j = i + 1; j < size; j++) {

        if (arr[i] == arr[j]) {

            duplicate = true;

            break;

        }

    }

    if (!duplicate) cout << arr[i] << " ";

}

cout << endl;

return 0;
```

```
Enter the size of the array: 4
Enter the elements of the array: 5 6 7 8
The unique array is: 5 6 7 8
```

```
...Program finished with exit code 0
Press ENTER to exit console.
```

Q4 (A)

```
#include <iostream>

using namespace std;

int main() {
    int size;
    cout << "Enter the size of the array: ";
    while (!(cin >> size) || size <= 0) {
        cin.clear();
        cin.ignore(10000, '\n');
        cout << "Invalid size! Enter again: ";
    }

    int arr[size];
    cout << "Enter the elements: ";
    for (int i = 0; i < size; i++) {
        while (!(cin >> arr[i])) {
```

```
    cin.clear();

    cin.ignore(10000, '\n');

    cout << "Invalid input! Enter again: ";

}

}

int start = 0, end = size - 1;

while (start < end) {

    int temp = arr[start];

    arr[start] = arr[end];

    arr[end] = temp;

    start++;

    end--;

}

cout << "Reversed array: ";

for (int i = 0; i < size; i++) cout << arr[i] << " ";

cout << endl;

return 0;

}
```

```
Enter the size of the array: 4
Enter the elements: 5 6 7 8
Reversed array: 8 7 6 5

...Program finished with exit code 0
Press ENTER to exit console.
```

(B)

```
#include <iostream>

using namespace std;

int main() {
    int row1, col1, row2, col2;
    cin >> row1 >> col1;
    cin >> row2 >> col2;

    if (col1 != row2) {
        cout << "Multiplication not possible";
        return 0;
    }

    int mat1[100][100], mat2[100][100], mat3[100][100];
    for (int i = 0; i < row1; i++) {
        for (int j = 0; j < col1; j++) {
            cin >> mat1[i][j];
        }
    }
}
```

```
    }

}

for (int i = 0; i < row2; i++) {

    for (int j = 0; j < col2; j++) {

        cin >> mat2[i][j];

    }

}

for (int i = 0; i < row1; i++) {

    for (int j = 0; j < col2; j++) {

        mat3[i][j] = 0;

        for (int k = 0; k < col1; k++) {

            mat3[i][j] += mat1[i][k] * mat2[k][j];

        }

    }

}

for (int i = 0; i < row1; i++) {

    for (int j = 0; j < col2; j++) {

        cout << mat3[i][j] << " ";

    }

    cout << "\n";

}

return 0;

}
```

```
2 3
3 2
1 2 3
4 5 6
7 8
9 10
11 12
58 64
139 154

...Program finished with exit code 0
Press ENTER to exit console.[]
```

(C)

```
#include <iostream>

using namespace std;

int main() {
    int rows, cols;
    cout << "Enter rows and columns: ";
    cin >> rows >> cols;
    int mat[100][100];
    cout << "Enter matrix elements:\n";
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            cin >> mat[i][j];
        }
    }
}
```

```

    }

}

cout << "Transpose of the matrix:\n";

for (int i = 0; i < cols; i++) {

    for (int j = 0; j < rows; j++) {

        cout << mat[j][i] << " ";

    }

    cout << "\n";

}

return 0;
}

```

```

Enter rows and columns: 2 3
Enter matrix elements:
1 2 3
4 5 6
Transpose of the matrix:
1 4
2 5
3 6

...Program finished with exit code 0
Press ENTER to exit console.█

```

Q5

```
#include <iostream>
```

```
using namespace std;
```

```
int main() {
```

```
    int rows, cols;
```

```
cout << "Enter number of rows: ";
cin >> rows;
cout << "Enter number of columns: ";
cin >> cols;

int arr[100][100];
cout << "Enter elements of the matrix:\n";
for (int i = 0; i < rows; i++) {
    for (int j = 0; j < cols; j++) {
        cin >> arr[i][j];
    }
}

cout << "Sum of each row:\n";
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 << " sum = " << rowSum << "\n";
}

cout << "Sum of each column:\n";
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 << " sum = " << colSum << "\n";  
  
}  
  
return 0;  
}
```

```
Enter number of rows: 2  
Enter number of columns: 3  
Enter elements of the matrix:  
1 2 3  
4 5 6  
Sum of each row:  
Row 1 sum = 6  
Row 2 sum = 15  
Sum of each column:  
Column 1 sum = 5  
Column 2 sum = 7  
Column 3 sum = 9  
  
...Program finished with exit code 0  
Press ENTER to exit console.
```