OPERATIONS

```
#include <iostream>
using namespace std;
const int MAX = 5;
class array
   {
      private:
            int arr[ MAX ];
      public:
            void insert (int pos, int num);
            void del (int pos);
            void reverse();
            void display();
            void search (int num);
   };
```

```
void array:: insert (int pos, int num)
   {
        int i;
        // shift elements to right
        for (i = MAX - 1; i >= pos; i--)
              arr[i] = arr[i - 1];
          arr[i] = num;
   }
```

```
void array:: del (int pos)
   {
    int i;
    // skip to the desired position
    for (i = pos; i < MAX; i++)
          arr[i - 1] = arr[i];
    arr[i - 1] = 0;
  }
```

Question 6: Write function code to delete one element from the array, if the array size is 5. [3 points]

451

```
void del(int Pos) {
for(int i = Pos; i<5; i++) {
arr[i-1] = arr[i];
arr[i-1] = 0;
}
```

Question 4: Write a function deletes the duplicated elements in the array data structure

461

```
void del_dup(){
for(int i=0; i<MAX; i++){
  for(int j=i; j<MAX; j++){
    if(arr[i]==arr[j]){int k;
      for(k=j; j<MAX-1; j++)
        arr[k] = arr[k+1];
      arr[k] = 0;
    }
  }
}
```

Question 6: Write a function to delete the duplicated elements in the array, if the array size is 10. [3 points]

452

```
void array::rem_dup () {
  for (int i=0;i<10;i++)
  {
    for (int j=i+1;j<10;j++)
    {
      if (a[i]==a[j]){
        int k;
        for(k=j; k<10-1; k++)
        {
          a[k]=a[k+1];
```

```
void array:: reverse()
{
    for (int i = 0; i < MAX / 2; i++)
    {
        int temp = arr[i];
        arr[i] = arr[MAX - 1 - i];
        arr[MAX - 1 - i] = temp;
    }
}
```

```
void array:: search (int num)
{
    int i;
    for (i = 0; i < MAX; i++)
    {
        if (arr[i] == num)
            cout << endl << "Element" << num<< "is at" << (i + 1) << "th position";
    }
    if (i == MAX)
        cout << endl << "Element" << num << "is absent";
}
```

```
void array:: display()
{
    cout << endl;
    for (int i = 0; i < MAX; i++)
    cout << arr[i] << "";
}
```

Question 5: Write a function code to display all elements in the array, if the array size is 8. *[2 points]*

451

```
void display() {
for(int i=0; i<8; i++){
  cout<<endl;
cout<<arr[i]<<" ";
  }
}
```

```
#include <iostream>
using namespace std;
const int MAX = 3;
class matrix
    {
      private:
           int mat[MAX][MAX];
      public:
           matrix(); void create();
           void display();
           void matadd (matrix &m1, matrix &m2);
           void matmul (matrix &m1, matrix &m2);
```

```
matrix:: matrix()
{
        for (int i = 0; i < MAX; i++)
        {
                for (int j = 0; j < MAX; j++)
                        mat[i][j] = 0;
        }
}
```

```
void matrix:: create()
{
    int n;
    for (int i = 0; i < MAX; i++)
    {
        for (int j = 0; j < MAX; j++)
        {
            cout << endl << "Enter the element:";
            cin >> n;
            mat[i][j] = n;
        }
    }
}
```

```
void matrix:: display()
{
        for (int i = 0; i < MAX; i++)
        {
                for (int j = 0; j < MAX; j++)
                        cout << mat[i][j] << "";
                cout << endl;
        }
}
```

Question 5: Write a function code to display elements of 4x4 matrix.

[3

452

```
void matrix:: display()
{
        for (int i = 0; i < 4; i++)
        {
                for (int j = 0; j < 4; j++)
                        cout << mat[i][j] << "";
                cout << endl;
        }
}
```

```
void matrix:: matadd (matrix &m1, matrix &m2)
{
        for (int i = 0; i < MAX; i++)
        {
                for (int j = 0; j < MAX; j++)
                        mat[i][j] = m1.mat[i][j] + m2.mat[i][j];
        }
}
```

```
void matrix:: matadd (matrix &m1, matrix &m2)
{
        int mat[3][3];
        for (int i = 0; i < 3; i++)
        {
          for(int j=0; j<3; j++){
            mat[i][j] = m1.mat[i][j] + m2.mat[i][j];}
        }
}
```

451

```
void matrix:: matmul (matrix &m1, matrix &m2)
{
    for (int k = 0; k < MAX; k++)
    {
        for (int i = 0; i < MAX; i++)
        {
            for (int j = 0; j < MAX; j++)
                mat[k][i] += m1.mat[k][j] * m2.mat[j][i];
        }
    }
}
```

```
void matrix:: transpose (matrix &m1)
{
    for (int i = 0; i < MAX; i++)
    {
        for (int j = 0; j < MAX; j++)
            mat[i][j] = m1.mat[j][i];
    }
}
```

Question 7: Write a function code to interchange the rows with corresponding columns of a matrix with size 4x4. [3 points]

451

```
void transpose(
Void matrix:: transpose (matrix &m1) {
int mat[4][4]
for(int i=0; i<4; i++){
    for(int j=0; j<4; j++){
    mat[i][j] = m1.mat[j][i]; }
}
```

462

```
void copyMatrix(matrix &m1) {
    for (int i = 0; i < 2; i++)
        for (int j = 0; j < 4; j++)
            Matrix[i][j] = m1.matrix[i][j];
}
```

```
#include
using namespace std;
class linklist
{
private:
// structure containing a data part //and link
part
      struct node
      {
            int data;
            node* link;
      } *p;
public:
      linklist();
      void append (int num);
      void addatbeg (int num);
      void addafter (int loc, int num);
      void display();
      int count();
      void del (int num);
      ~linklist();
};
```

```
void linklist:: append (int num)
{
        node *temp, *r;
        // if the list is empty, create first node
        if (p == NULL)
        {
                temp = new node;
                temp -> data = num;
                temp -> link = NULL;
                p = temp;
        }
        else
        {
                // go to last node
                temp = p;
                while (temp -> link != NULL)
                        temp = temp -> link;
                // add node at the end
                r = new node;
                r -> data = num;
                r -> link = NULL;
                temp -> link = r;
        }
}
```

```
void linklist:: addatbeg (int num)
{
// It adds a new node at the //beginning
of the linked list
        node *temp;
        temp = new node;
        temp -> data = num;
        temp -> link = p;
        p = temp;
}
```

```
void linklist:: addafter (int loc, int num)
{
// It adds a new node after the specified //number
of nodes in the linked list.
      node *temp, *m;
      temp = p;
      // skip to desired portion
      for (int i = 0; i < loc - 1; i++)
      {
            temp = temp -> link;
            // if end of linked list is //encountered
            if (temp == NULL)
            {
                  cout << "\n There are less than" <<
                  loc
                  << "elements in list" << endl;
                  return;
            }
      }
      // insert new node
      m = new node;
      m -> data = num;
      m -> link = temp -> link;
      temp -> link = m;
}
```

```
void linklist:: display()
{
        node *temp = p;
        // traverse the entire linked list
        while (temp != NULL)
        {
                cout << temp -> data << “”;
                temp = temp -> link;
        }
        cout << endl;
}
```

```
int linklist:: count()
{
        int c = 0;
        node *temp = p;
        // traverse the entire linked list
        while (temp != NULL)
        {
                c++;
                temp = temp -> link;
        }
        return c;
}
```

462

```
int countOddNumbers() {
    int count = 0;
Node* Temp =p;
    while (temp!=null) {
        if (temp->data % 2 ==1)
            count++;
        Temp = temp->link;
    }
    return count;
}
```