

(1)

```
void Reverse_Array (int * Arr , int Size)
{
    for (int i = 0; i < Size/2; i++)
    {
        int temp = Arr[i];
        Arr[i] = Arr[Size -1 -i];
        Arr[Size -1 -i] = temp;
    }
}

void main()
{
    // Reverse Array
    int Size;
    cout << "Enter Size of Arrays: ";
    cin >> Size;
    int* Arr1 = new int [Size];
    cout << "Enter Values of Array 1 :";
    for(int i=0;i<Size;i++)
    {
        cin >> Arr1[i];
    }
    Reverse_Array(Arr1,Size);
    cout << "Array After Reverse :";
    for (int i = 0; i < Size; i++)
    {
        cout << Arr1[i] << " ";
    }
    cout << endl;

    delete[] Arr1;
}
```

(2)

```
int Max_Odd (int* Arr, int Size)
{
    int MaxOdd = -1;
    for (int i = 0; i < Size; i++)
    {
        if(Arr[i]%2!=0)
        {
            if(MaxOdd == -1 || MaxOdd < Arr[i])
            {
                MaxOdd = Arr[i];
            }
        }
    }
    return MaxOdd;
}

void main()
{
    int Size;
    cout << "Enter Size of Arrays: ";
    cin >> Size;
    int* Arr1 = new int [Size];
    cout << "Enter Values of Array 1 :";
    for(int i=0;i<Size;i++)
    {
        cin >> Arr1[i];
    }
    int MaxOddNumber = Max_Odd(Arr1 , Size);
    cout << "Maximum Odd Number in The Array : "<<MaxOddNumber << endl;
    delete[] Arr1;
}
```

(3)

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

```
void TrimExtraSpace(string &str)//[7]
{
    int size = str.length()-1;
    for (int i = 0; i < size; i++)//[1]
    {
        if (str[i] == ' ' && str[i + 1] == ' ')//[1]
        {
            for (int j = i + 1; j < size; j++)//[1]
            {
                char temp = str[j];//[1]
                str[j] = str[j + 1];//[1]
                str[j + 1] = temp;//[1]
            }
            i--;
            size--;
        }
    }
}
```

```
void main()//[3]
{
    string str;//[1]

    cout << "Enter any string: ";
    getline(cin, str);//[1/2]

    TrimExtraSpace(str);//[1]

    cout << "String after trimming: " << str << endl;//[1/2]
}
```

(4)

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

void input(string *arr, int size);
void SearchAll(string *arr, int size, string Keyword);

void main()
{
    int size;
    string *a;
    string key;
    cout<< "enter Size\n";
    cin >> size;
    a = new string[size];
    input(a, size);
    cout << "Enter the keyword to search for :\n";
    cin>> key;
    SearchAll(a, size, key);
}

void input(string *arr, int size)
{
    cout << "Enter the "<<size<<" strings\n ";
    for (int i = 0; i < size; i++)
        cin >> arr[i];
}
//All word that have er:
void SearchAll(string *arr, int size, string Keyword)
{
    cout << "All word that have " << Keyword << endl;
    for (int i = 0; i < size; i++)
    {
        if (arr[i].find(Keyword) != string::npos)
            cout << arr[i] << endl;
    }
}
```

(5)

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

bool isSparse(int** arr, int row, int cols);
void input(int** arr, int row, int cols);

void main()
{
    int length, width;
    char choice;

    do
    {
        cout << "Enter the length: ";
        cin >> length;
        cout << "Enter the width: ";
        cin >> width;

        int **arr = new int*[length];
        for (int i = 0; i < length; i++)
        {
            arr[i] = new int[width];
        }

        input(arr, length, width);

        if (isSparse(arr, length, width))
            cout << "It's a sparse!!" << endl;
        else
            cout << "Not a sparse!!" << endl;

        for (int i = 0; i < length; i++)
        {
            delete[]arr[i];
        }
        delete[]arr;

        cout << "Do you want to continue (y/n)?"<<endl;
        cin >> choice;
    } while (choice == 'y' || choice == 'Y');
```

```
}
```

```
bool isSparse(int** arr, int row, int cols)
```

```
{
```

```
    int count = 0;
```

```
    for (int i = 0; i < row; i++)
```

```
    {
```

```
        for (int j = 0; j < cols; j++)
```

```
        {
```

```
            if (arr[i][j] == 0)
```

```
            {
```

```
                count++;
```

```
            }
```

```
        }
```

```
    }
```

```
    if (count >= (row*cols) / 2.0)
```

```
        return true;
```

```
    return false;
```

```
}
```

```
void input(int** arr, int rows, int cols)
```

```
{
```

```
    cout << "Enter the matrix: " << endl;
```

```
    for (int i = 0; i < rows; i++)
```

```
    {
```

```
        for (int j = 0; j < cols; j++)
```

```
        {
```

```
            cin >> arr[i][j];
```

```
        }
```

```
    }
```

```
}
```

(6)

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

```
void input(int** arr, int rows, int cols);
int Sumdiagonal(int** arr, int row, int cols);
```

```
void main()
```

```
{
    int length;
    char choice;
    do
    {
        cout << "Enter the size: ";
        cin >> length;
        int **arr = new int*[length];
        for (int i = 0; i < length; i++)
        {
            arr[i] = new int[length];
        }
        input(arr, length, length);

        cout << "Sum of diagonal elements= " << Sumdiagonal(arr, length, length) <<
endl;

        for (int i = 0; i < length; i++)
        {
            delete[]arr[i];
        }
        delete[]arr;
        cout << "Do you want to continue (y/n)?"<<endl;
        cin >> choice;
    }while (choice == 'y' || choice == 'Y');
}
```

```
void input(int** arr, int rows, int cols)
{
    cout << "Enter the matrix: " << endl;
    for (int i = 0; i < rows; i++)
    {
        for (int j = 0; j < cols; j++)
        {
```

```
        cin >> arr[i][j];
    }
}
```

```
int Sumdiagonal(int** arr, int row, int cols)
{
    int sum = 0;
    for (int i = 0; i < row; i++)
    {
        sum += arr[row - 1 - i][i];
    }
    return sum;
}
```



(7)

```
#include <iostream>
using namespace std;
bool check_uniqueness(int * arr, int size)
{
    bool unique = true;
    bool flag = false;
    for (int i = 0; i < size; i++)
    {
        for (int j = i + 1; j < size; j++)
        {
            if (arr[i] == arr[j])
            {
                unique = false;
                flag = true;
                break;
            }
        }
        if (flag == true)
            break;
    }
    if (unique == true)
        return true;
    else
        return false;
}

int main()
{
    int size;
    char choice;
    do
    {
        cout << "enter size" << endl;
        cin >> size;
        int *arr = new int[size];
        cout << "enter elements" << endl;
        for (int i = 0; i < size; i++)
        {
            cin >> arr[i];
        }
        bool res = check_uniqueness(arr, size);
```

```
        if (res == true)
            cout << "All elements are unique." << endl;
        else
            cout << "Elements are not unique." << endl;
        delete[]arr;
        cout << "Do you want to continue (y/n)?" << endl;
        cin >> choice;
    } while (choice == 'y' || choice == 'Y');

return 0;
}
```

(8)

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

```
int main()
{
    int size1;
    int size2;
    cout << "please enter the size of divisors list" << endl;
    cin >> size1;
    int *divisors = new int[size1];
    cout << "enter the divisors list" << endl;
    for (int i = 0; i < size1; i++)
    {
        cin >> divisors[i];
    }
    cout << "enter size of values list" << endl;
    cin >> size2;
    int *values = new int[size2];
    cout << "enter the values list" << endl;
    for (int i = 0; i < size2; i++)
    {
        cin >> values[i];
    }

    cout << "-----result-----" << endl;
    int *result = new int[size2];
    for (int i = 0; i < size1; i++)
    {
        int z = 0;
        for (int j = 0; j < size2; j++)
        {
            if (values[j] % divisors[i] == 0)
            {
                result[z] = values[j];
                z++;
            }

            cout << "the list ";
        }
        for (int k = 0; k < z; k++)
        {
```

```
        cout << result[k] << " - ";
    }
    cout << "is divided by " << divisors[i] << endl;

}

delete[] values;
delete[] divisors;
delete[] result;
return 0;
}
```

(9)

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

```
struct Lab
{
    int ID, MaxStudents;
    int length, with;
};
```

```
void calculateStudents(Lab *arr, int n);
void findLabs(Lab *arr, int n, int studs);
```

```
void main()
{
    int size, Nstuds;
    Lab *L;
    cout << "Enter the number of Lab:\n";
    cin >> size;
    L = new Lab[size];
    for (int i = 0; i < size; i++)
    {
        cout << "Enter the ID:\n";
        cin >> L[i].ID;
        cout << "Enter the dimension length :\n";
        cin >> L[i].length;
        cout << "Enter the dimension width : \n";
        cin >> L[i].with;
    }
    calculateStudents(L, size);
    cout << "Enter the Number of students: ";
    cin >> Nstuds;
    findLabs(L, size, Nstuds);
}
```

```
void calculateStudents(Lab *arr, int n)
{
    int Area;
    for (int i = 0; i < n; i++)
    {
        Area = arr[i].length* arr[i].with;
        arr[i].MaxStudents = Area * 5;//10/2
    }
}
```

```

    }
}

void findLabs(Lab *arr, int n, int studs)
{
    cout << "The Lab with students higher than or equal to the students:\n";

    for (int i = 0; i < n; i++)
    {
        if (arr[i].MaxStudents >= studs)
            cout << "Lab ID : " << arr[i].ID << " has Max Students : " <<
arr[i].MaxStudents << endl;
    }
}

```

(10)

```
#include <iostream>
#include<string>
using namespace std;
struct student
{
    string name;
    int id;
    string address;
};
int main()
{
    int num;
    cout << "Enter the number of students" << endl;
    cin >> num;
    student *stds = new student[num];
    for (int i = 0; i<num; i++)
    {
        cout << "student #" << i + 1 << ": " << endl;
        cout << "Name: ";
        cin >> stds[i].name;
        cout << "id: ";
        cin >> stds[i].id;
        cout << "address #: ";
        cin >> stds[i].address;
    }
    cout << "----- result-----" << endl;
    for (int i = 0; i<num; i++)
    {
        for (int j = i + 1; j<num; j++)
        {
            if (stds[i].address == stds[j].address)
            {
                cout << stds[i].name << " and " << stds[j].name << " have the
same address " << stds[i].address << endl;
            }
        }
    }

    delete[]stds;
    return 0;
}
```