1>

```cpp
};
string bSearch(struct Array arr1,int key){
    int i,j,mid;
    i=0;
    j=arr1.length-1;
    while(i<=j){
        mid = (i+j)/2;
        if(key<arr1.A[mid]){
            j=mid-1;
        }
        else if(key>arr1.A[mid]){
            i=mid+1;
        }
        else{
            return "key found succesfully";
        }
    }
    return "key not found";
}
int main(){
    struct Array arr1={{1,2,4,7,9,14,17,23},10,8};
    cout << bSearch(arr1,9);
    return 0;
}
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
...
key found succesfully
```

2>

AQ2.cpp > main()

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

void sortArray(int numbers[], int size) {
    for (int pass = 0; pass < size - 1; pass++) {
        for (int i=0; i< size-pass-1; i++) {
            if (numbers[i] > numbers[i + 1]) {
                int temp = numbers[i];
                numbers[i] = numbers[i + 1];
                numbers[i+1] = temp;
            }
        }
    }
}

int main() {
    int data[] = {66, 33,22, 11,55, 77, 99};
    int length = sizeof(data) / sizeof(data[0]);
    sortArray(data, length);
    cout << "sorted array: ";
    for (int i = 0; i < length; i++) {
        cout << data[i] << " ";
    }
    cout << endl;

```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

sorted array: 11 22 33 55 66 77 99
```

3>

```cpp
#include<iostream>
using namespace std;
string linear_search(int arr[],int key,int length){
    int i=0;
    while(i<length){
        if(arr[i]==key){
            return "key succesfully found ";
        }
        i++;
    }
    return "key not found" ;
}
int main(){
    int arr[]={2,4,6,9,13,18,25,55,56};
    int n = sizeof(arr)/sizeof(arr[0]);
    cout << linear_search(arr,56,n);
    return 0;
}
```

3 b>

```cpp
#include<iostream>
using namespace std;
int findMissingNumber(int arr[], int size) {
    int l = 0;
    int r = size - 1;
    if (size == 0) return 1;
    if (arr[0] != 1) return 1;
    if (arr[size - 1] != size + 1) return size + 1;

    while (l <= r) {
        int mid = l + (r-l) / 2;
                if (arr[mid] != mid + 1) {
            if (mid == 0 || arr[mid - 1] == mid) {
                return mid + 1;
            }
            r = mid - 1;
        } else {
            l = mid + 1;
        }
    }

    return -1;
}
int main(){
    int arr[]={1,2,3,4,5,6,7};// n = 8, size = 7, missing 8
    int n = sizeof(arr)/sizeof(arr[0]);
    int missing = findMissingNumber(arr, n);
    cout << "Missing number: " << missing << endl;
    return 0;
```

```
Missing number: 8
```

4 and 5 are same as assignment 1

6>

```
class Sparse {
private:
    int r[20], c[20], v[20], n=4, s;
public:
    Sparse(int sz, int e[][3]) : s(sz) {
        for(int i=0; i<s; i++) {
            r[i]=e[i][0];
            c[i]=e[i][1];
            v[i]=e[i][2];
        }
    }
    void print() {
        for(int i=0; i<n; i++) {
            for(int j=0; j<n; j++) {
                int val=0;
                for(int k=0; k<s; k++)
                    if(r[k]==i && c[k]==j) {val=v[k]; break;}
                cout<<val<<" ";
            }
            cout<<endl;
        }
    }
    Sparse transpose() {
        int tr[20], tc[20], tv[20];
        for(int i=0; i<s; i++) {
            tr[i]=c[i];
            tc[i]=r[i];
            tv[i]=v[i];
        }
        int e[20][3];
        for(int i=0; i<s; i++) {
            e[i][0]=tr[i];
            e[i][1]=tc[i];
            e[i][2]=tv[i];
        }
        return Sparse(s, e);
    }
```

```
Sparse add(Sparse &b) {
    int tr[40], tc[40], tv[40], k=0, i=0, j=0;
    while(i<s && j<b.s) {
        if(r[i]<b.r[j] || (r[i]==b.r[j] && c[i]<b.c[j])) {
            tr[k]=r[i]; tc[k]=c[i]; tv[k]=v[i]; i++;
        }
        else if(r[i]>b.r[j] || (r[i]==b.r[j] && c[i]>b.c[j])) {
            tr[k]=b.r[j]; tc[k]=b.c[j]; tv[k]=b.v[j]; j++;
        }
        else {
            tr[k]=r[i]; tc[k]=c[i]; tv[k]=v[i]+b.v[j]; i++; j++;
        }
        k++;
    }
    while(i<s) {tr[k]=r[i]; tc[k]=c[i]; tv[k]=v[i]; i++; k++;}
    while(j<b.s) {tr[k]=b.r[j]; tc[k]=b.c[j]; tv[k]=b.v[j]; j++; k++;}
    int e[40][3];
    for(int m=0; m<k; m++) {
        e[m][0]=tr[m]; e[m][1]=tc[m]; e[m][2]=tv[m];
    }
    return Sparse(k, e);
}
```

```
    Sparse multiply(Sparse &b) {
        int tr[40], tc[40], tv[40], k=0;
        for(int i=0; i<s; i++) {
            for(int j=0; j<b.s; j++) {
                if(c[i]==b.r[j]) {
                    int row=r[i], col=b.c[j], prod=v[i]*b.v[j], f=-1;
                    for(int m=0; m<k; m++)
                        if(tr[m]==row && tc[m]==col) {f=m; break;}
                    if(f!=-1) tv[f]+=prod;
                    else {
                        tr[k]=row; tc[k]=col; tv[k]=prod; k++;
                    }
                }
            }
        }
        int e[40][3];
        for(int m=0; m<k; m++) {
            e[m][0]=tr[m]; e[m][1]=tc[m]; e[m][2]=tv[m];
        }
        return Sparse(k, e);
    }
};

int main() {
    int a[4][3]={{0,0,1},{1,2,2},{2,1,3},{3,3,4}};
    Sparse A(4,a);
    int b[4][3]={{0,1,5},{1,1,6},{2,2,7},{3,0,8}};
    Sparse B(4,b);
    cout<<"Matrix A:\n"; A.print();
    cout<<"\nMatrix B:\n"; B.print();
    cout<<"\nTranspose A:\n"; A.transpose().print();
    cout<<"\nA + B:\n"; A.add(B).print();
    cout<<"\nA * B:\n"; A.multiply(B).print();
    return 0;
}
```

```
Matrix A:
1 0 0 0
0 0 2 0
0 3 0 0
0 0 0 4

Matrix B:
0 5 0 0
0 6 0 0
0 0 7 0
8 0 0 0

Transpose A:
1 0 0 0
0 0 3 0
0 2 0 0
0 0 0 4

A + B:
1 5 0 0
0 6 2 0
0 3 7 0
8 0 0 4

A * B:
0 5 0 0
0 0 14 0
0 18 0 0
32 0 0 0
```

7>

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

int countInversions(int a[], int n) {
    int c = 0;
    for(int i=0; i<n-1; i++)
        for(int j=i+1; j<n; j++)
            if(a[i]>a[j]) c++;
    return c;
}

int main() {
    int a[5] = {5, 2, 4, 1, 3};
    int n = 5;
    cout<<"Array: ";
    for(int i=0; i<n; i++) cout<<a[i]<<" ";
    cout<<"\nInversions: "<<countInversions(a, n);
    return 0;
}
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\HP\OneDrive\Desktop\2ND_YEAR\DSA_ASSIGNMENTS\assignment-2-arrays-AmitBishnoi2005>  & 'c:\Users\HP\.vscode\extensions\ms-vscode.cp
ools-1.28.3-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-10gh5d4s.2p4' '--stdout=Microsoft-MIEngine-O
-4hfuqq0x.fkg' '--stderr=Microsoft-MIEngine-Error-czxe5pyo.ask' '--pid=Microsoft-MIEngine-Pid-rl01d4pb.etb' '--dbgExe=C:\msys64\ucrt64\bin\gd
exe' '--interpreter=mi'
Array: 5 2 4 1 3
Inversions: 7
PS C:\Users\HP\OneDrive\Desktop\2ND_YEAR\DSA_ASSIGNMENTS\assignment-2-arrays-AmitBishnoi2005>
```

8>

AQ8.cpp > main()

```cpp
#include <iostream>
using namespace std;
int main() {
    int arr[] = {1, 2, 3, 6, 3, 1, 5, 7, 22, 13, 456, 32, 45, 2};
    int n = sizeof(arr) / sizeof(arr[0]);
    int distinct_count = 0;

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

        for (int j = 0; j < i; j++) {
            if (arr[i] == arr[j]) {
                is_duplicate = true;
                break;
            }
        }
        if (!is_duplicate) {
            distinct_count++;
        }
    }

    cout << "Total no of distinct elements: " << distinct_count << endl;

    return 0;
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\HP\OneDrive\Desktop\2ND_YEAR\DSA_ASSIGNMENTS\assignment-2-arrays-AmitBishnoi2005>  & 'c:\Users\HP\.vscode\extensions\ms-vscode.c
ools-1.28.3-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-4jzoh2tx.l14' '--stdout=Microsoft-MIEngine-
-udujqyy0.bti' '--stderr=Microsoft-MIEngine-Error-njdriv3o.vlv' '--pid=Microsoft-MIEngine-Pid-t2sxjby5.zxa' '--dbgExe=C:\msys64\ucrt64\bin\g
exe' '--interpreter=mi'
Total no of distinct elements: 11
PS C:\Users\HP\OneDrive\Desktop\2ND_YEAR\DSA_ASSIGNMENTS\assignment-2-arrays-AmitBishnoi2005>
```