

- 1). Write a program in C for, In Array 1-100 numbers are stored, one number is missing how do you find it.

```

int main() {
    int i, n, sum=0, missing;
    cout << "Enter size of array ";
    cin >> n;
    int arr[n-1];
    cout << "Enter values ";
    for (i=0; i<n-1; i++) {
        cin >> arr[i];
        sum = sum + arr[i];
    }
    missing = (n*(n+1))/2 - sum;
    cout << "Missing number is " << missing;
}

```

- 2). Program to print duplicate numbers with frequency in an array.

```

int main() {
    int size, i, num, occr=0;
    cout << "Enter size ";
    cin >> size;
    for (i=0; i<size; i++) {
        cin >> arr[i];
    }
    cout << "Enter array item to know ";
    cin >> num;
    for (i=0; i<size; i++) {
        if (arr[i] == num) {
            occr++;
        }
    }
    cout << "num << occurred " << occr << "times ";
}

```

3.

3). Program to print duplicate elements of an array.

```
int main() {
    for (int i=0; i<length; i++) {
        for (int j=i+1; j<length; j++) {
            if (arr[i]==arr[j]) {
                cout << arr[j];
            }
        }
    }
}
```

4). Program to print first repeating element in an array

```
int main() {
    for (int i=0; i<n; i++) {
        for (int j=i+1; j<n; j++) {
            if (arr[i]==arr[j]) {
                cout << "First Repeating integer
is" << arr[i];
                return 0;
            }
        }
    }
    cout << "No integer repeated";
    return 0;
}
```

5). Program to remove duplicate elements from Array.

```
int main() {
```

```
    int arr[50], i, j, k, count = 0, dup[50], number;
```

```
    cout << "Enter size of array";
```

```
    cin >> number;
```

```
    cout << "Enter elements of array";
```

```
    for (i=0; i<number; i++) {
```

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

```
        dup[i] = -1;
```

```
    for (i=0; i<number; i++) {
```

```
        for (j=i+1; j<number; j++) {
```

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

```
                for (k=j; k<number; k++) {
```

```
                    arr[k] = arr[k+1];
```

```
}
```

```
        j--;
```

```
}
```

```
number--;
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

<div data-bbox="50

6) Program for, Given two arrays 1,2,3,4,5 and 2,3,1,0,5 , find which number is not present in the second array.

```

int main(){
    int array1[] = {1,2,3,4,5};
    int array2[] = {2,3,1,0,5};
    int check = 0;
    int num = 0;
    for (int i = 0; i < 5; i++) {
        for (int j = 0; j < 5; j++) {
            if (array1[i] == array2[j]) {
                check = 1;
            }
        }
        if (check == 0) {
            num = array1[i];
            check = 0;
        }
    }
}

```

cout << "The number present in first array but
not in second is " << num;

7). Program to check two arrays is equal in size or not.

```
int main()
{
    int arr1[] = {1, 2, 3, 4, 5};
    int arr2[] = {2, 3, 4, 5, 6};
    int size1 = sizeof(arr1) / sizeof(arr1[0]);
    int size2 = sizeof(arr2) / sizeof(arr2[0]);
    if (size1 == size2)
        cout << "Same";
    else
        cout << "Not same";
}
```

8) Program to find largest and smallest number in array.

```
int main()
{
    int arr[], n, large, small;
    cout << "Enter no. of elements";
    cin >> n;
    cout << "Enter array elements";
    for (i = 0; i < n; i++)
        cin >> arr[i];
    large = small = arr[0];
    for (i = 1; i < n; i++)
    {
        if (arr[i] > large)
            large = arr[i];
        if (arr[i] < small)
            small = arr[i];
    }
    cout << small << endl;
    cout << large << endl;
}
```

9). Program to print array in reverse order.

```

int main()
{
    int n,i;
    cout << "Enter size of an array";
    cin >> n;
    int arr[n];
    for (i=0;i<n;i++)
        cin >> arr[i];
    }
    cout << "Reverse array is ";
    for (i=n-1;i>=0;i--)
    {
        cout << arr[i];
    }
    return 0;
}

```

10) Program to insert an element at end of an array.

```

void main()
{
    int position, i, n, value, ch;
    arr[100];
    cin >> position >> n;
    arr[n];
    for (i=0;i<n;i++)
        cin >> arr[i];
    cout << "Enter value to be added at the end";
    cin >> value;
    arr[n] = value;
    for (i=0; i<n+1; i++)
        cout << arr[i];
}

```

11) Program to insert element at a given location in array.

```

void main() {
    int loc, i, n, value, ch;
    cin >> n;
    int arr[n];
    for (i = 0; i < n; i++)
        cin >> arr[i];
    cout << "Enter number to insert";
    cin >> value;
    cout << "Enter location to insert as new element";
    cin >> loc;
    for (i = n - 1; i >= loc - 1; i--)
        arr[i + 1] = arr[i];
    arr[loc - 1] = value;
    for (i = 0; i < n; i++)
        cout << arr[i];
}

```

12) Program to delete element at end of array.

```

void main() {
    int position, i, n, value, ch;
    cin >> n;
    int arr[n];
    for (i = 0; i < n; i++)
        cin >> arr[i];
    value = arr[n - 1];
    n = n - 1;
    for (i = 0; i < n; i++)
        cout << arr[i];
}

```

13). Program to delete given element from array.

```

int main()
{
    int i, size, value, j, temp;
    cout << "Enter size of array";
    cin >> size;
    int arr[size];
    temp = size;
    for (i=0; i<size; i++) {
        cin >> arr[i];
    }
    cout << "Enter element to delete";
    cin >> value;
    for (i=0; i<size; i++) {
        if (arr[i] == value) {
            for (j=i; j<(size-1); j++) {
                arr[j] = arr[j+1];
            }
            size--;
            i--;
        }
    }
    if (temp == size)
        cout << "No element found in array";
    exit(0);
}

cout << "Elements after deleting" -;
for (i=0; i<size; i++) {
    cout << arr[i];
}
return 0;
}

```

14). Program to delete element from array at given index.

```

void main() {
    int loc, i, size, value, ch;
    cin >> size;
    int arr[size];
    for (i=0; i<size; i++)
        cin >> arr[i];
    cout << "Enter index to delete an element";
    cin >> loc;
    if (loc <= size-1) {
        for (i=loc; i<size-1; i++) {
            arr[i] = arr[i+1];
        }
        size--;
    } else {
        cout << "Index not available";
        exit(0);
    }
    cout << "After Deletion Array is";
    for (i=0; i<size; i++)
        cout << arr[i];
}

```

3.

15). Program to perform left rotation of array elements by two positions.

Ex: $a = \{1, 2, 3, 4, 5\}$. first time left rotation = $\{2, 3, 4, 5, 1\}$
 second time left rotation = $\{3, 4, 5, 1, 2\}$

```
int main() {
    int i, size, j, k, temp;
    cin >> size;
    int arr[size];
    for (i=0; i<size; i++) {
        cin >> arr[i];
    }
    for (i=0; i<2; i++) {
        temp = arr[0];
        for (j=0; j<size-1; j++) {
            arr[j] = arr[j+1];
        }
        arr[size-1] = temp;
    }
}
```

cout << "array after two time left rotation : ";
 for (i=0; i<size; i++) {
 cout << arr[i];
 }
}

} octomos

BONAROOPA						
S	M	T	W	T	F	S
Page No.:	Data:	11				

16) Program to print / perform right rotation in array by k positions.

ex: $a = \{1, 2, 3, 4, 5\}$. \rightarrow 1st $\rightarrow \{5, 1, 2, 3, 4\}$ \rightarrow 2nd $\rightarrow \{4, 5, 1, 2, 3\}$

```

int main() {
    int size, j, k, temp; // size - size of array
    cin >> size; // reading size of array
    int arr[size]; // size of array
    for (i=0; i<size; i++) // reading
        cin >> arr[i];
    for (i=0; i<2; i++) { // for two times
        temp = arr[size-1];
        for (j=size-1; j>0; j--) {
            arr[j] = arr[j-1];
        }
        arr[0] = temp; // arr[0] = arr[size-1]
    }
    cout << "After two times right rotation";
    for (i=0; i<size; i++)
        cout << arr[i];
    return 0;
}

```

13). Program to merge two arrays.

```

int main()
    int size1, size2, newSize, i, j;
    Read two arrays; arr1 & arr2;

    newSize = size1 + size2;
    int newArr[newSize];
    for (i=0; i<size1; i++) {
        newArr[i] = arr1[i];
    }
    for (i=0, j=size1; j<newSize & i<size2; i++, j++) {
        newArr[j] = arr2[i];
    }
    cout << "New Array after merging";
    for (i=0; i<newSize; i++) {
        cout << newArr[i];
    }
    return 0;
}

```

18) Program to find highest frequency element in array.

```

int main()
{
    int size, i, j, count = 0, maxCount = 0, maxElement, occurrence;
    cin >> size;
    for (i = 0; i < size; i++)
        cin >> arr[i];
    for (i = 0; i < size; i++) {
        count = 1;
        for (j = i + 1; j < size; j++) {
            if (arr[j] == arr[i]) {
                count++;
                if (count > maxCount) {
                    maxElement = arr[j];
                    maxCount = count;
                }
            }
        }
    }
    cout << "Maximum frequency element" << maxElement
        << "occurrence = " << maxCount;
    return 0;
}

```

19). Sort first half in ascending order and second half in descending.

```

int main(){
    int a[] = {3, 2, 4, 1, 10, 30, 40, 20};
    int n = sizeof(a) / sizeof(a[0]);
    int temp;
    for (int i=0; i<n-1; i++) {
        for (int j=0; j<n/2; j++) {
            if (a[j] > a[j+1]) {
                temp = a[j];
                a[j] = a[j+1];
                a[j+1] = temp;
            }
        }
        for (int j=n/2; j<n-1; j++) {
            if (a[j] < a[j+1]) {
                temp = a[j];
                a[j] = a[j+1];
                a[j+1] = temp;
            }
        }
        cout << a[i] << " ";
    }
    return 0;
}

```

20). Program to print repeated elements only 1st time.

Ex: 1 1 2 2 3. \Rightarrow O/p \rightarrow 1 2 3.

int main() {

 int count = 0;

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

 count = 0;

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

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

 count = 1;

 }

 if (count == 0) {

 cout << arr[i] << " ";

}

}

return 0;

}.

21). Program to count and print distinct elements from an array.

```

int main()
{
    int count = 0;
    for (int i = 0; i < n; i++) {
        int j;
        for (j = 0; j < i; j++) {
            if (array[i] == array[j])
                break;
            if (i == j)
                cout << array[i] << " ";
            count++;
        }
    }
    cout << endl << count;
    return 0;
}
  
```

Q: Zeros to end of an array.

```

int j = 0;
for (int i = 0; i < n; i++) {
    if (a[i] == 0) {
        swap(a[j], a[i]);
        j++;
    }
}
  
```

```

for (int i = 0; i < n; i++)
    cout << a[i];
  
```

22). Repeating elements in an array.

```
int main() {
    for (i=0; i<n; i++) {
        for (j=i+1; j<n; j++) {
            if (a[i] == a[j]) {
                cout << a[i];
            }
        }
    }
}
```

23). Program to print non-repeated elements of an array.

```
int main() {
    for (i=0; i<n; i++) {
        for (j=0; j<n; j++) {
            if (a[i] == a[j] && i != j) {
                break;
            }
        }
        if (j == n) {
            cout << a[i];
        }
    }
    cout << endl;
    return 0;
}
```

24). Program to remove duplicate elements from an array.

```
int removeDuplicates(int arr[], n) {
    int i=0;
    for (int j=1; j<n; j++) {
        if (arr[i] == arr[j]) {
            i++;
            arr[i] = arr[j];
        }
    }
    return i+1;
}
```

```
int main() {
    int arr[10];
    cin >> arr[0];
    int k = removeDuplicates(arr);
    for (i=0; i<k; i++) {
        cout << arr[i] << " ";
    }
}
```

! OR:

```
int n; cin >> n;
for (i=0; i<n; i++) {
    cin >> arr[i];
}

set<int> s;
for (i=0; i<n; i++) {
    s.insert(arr[i]);
}

int it;
for (it=s.begin(); it!=s.end(); it++)
    cout << *it;
```

Programs on Matrix

SONA ROOPA
S M T W T F S
Page No.: Date 30/4/12

1). Add two matrix.

```
int main()\n{\n    int i, j, rows, columns, a[10][10], b[10][10];\n    int arr[10][10];\n\n    cout << "Enter rows and columns: ";\n    cin >> rows >> columns;\n\n    cout << "Enter first matrix elements: ";\n    for (i=0; i<rows; i++) {\n        for (j=0; j<columns; j++) {\n            cin >> a[i][j];\n        }\n    }\n\n    cout << "Enter second matrix elements: ";\n    for (i=0; i<rows; i++) {\n        for (j=0; j<columns; j++) {\n            cin >> b[i][j];\n        }\n    }\n\n    for (i=0; i<rows; i++) {\n        for (j=0; j<columns; j++) {\n            arr[i][j] = a[i][j] + b[i][j];\n        }\n    }\n\n    cout << "Sum is: ";\n    for (i=0; i<rows; i++) {\n        for (j=0; j<columns; j++) {\n            cout << arr[i][j] << " ";\n        }\n    }\n}
```

3.

Signature

27. Check two matrices are equal or not.

isEqual = 1;

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

```
    for (j = 0; j < columns; j++) {
```

```
        if (a[i][j] != b[i][j]) {
```

```
            isEqual = 0;
```

```
            break;
```

}

```
if (isEqual == 1) {
```

```
    cout << "Matrix a is equal to Matrix b";
```

```
} else {
```

```
    cout << "Matrices are not equal";
```

}

```
return 0;
```

.

QUESTION 28
Program to find transpose of matrix.

3). Program to check matrix is an identity matrix or not.

```
int main() {
```

```
    int flag = 1;
```

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

```
        for (j=0; j<column; j++) {
```

```
            if (a[i][j] != 1 && a[i][j] != 0) {
```

```
                flag = 0;
```

```
                condition found break;
```

```
            } else if (a[i][j] == 1) {
```

```
}
```

```
}
```

if (flag == 1)

```
cout << "Matrix is Identity matrix";
```

```
else cout << "Matrix is not an Identity matrix";
```

```
}
```

5

Program: To find LCM of two numbers

Two numbers are taken as input

Print the LCM of two numbers

Input: 12 18 Output: 36

Input: 15 25 Output: 75

Input: 10 20 Output: 20

Input: 15 25 Output: 75

Input: 10 20 Output: 20

Input: 15 25 Output: 75

Input: 10 20 Output: 20

Input: 15 25 Output: 75

Signature

4). Program to interchange diagonals of a matrix.

```

int main()
{
    int temp;
    if (rows == columns) {
        for (rows = 0; i = 0; i < rows; i++) {
            temp = arr[i][i];
            arr[i][i] = arr[i][j];
            arr[i][j] = temp;
        }
    }
    cout << "Enter no. of rows and columns ";
    cin >> i >> j;

    if (i == j) {
        cout << "Enter matrix elements ";
        for (rows = 0; rows < i; rows++) {
            for (columns = 0; columns < j; columns++) {
                cin >> arr[rows][columns];
            }
        }
        for (rows = 0; rows < i; rows++) {
            temp = arr[rows][rows];
            arr[rows][rows] = arr[rows][i - rows - 1];
            arr[rows][i - rows - 1] = temp;
        }
    }

    cout << endl;
    cout << "Please Enter square matrix " >> j;
}
    
```

5). Lower Triangle Matrix. Ex: $\begin{bmatrix} 10 & 20 \\ 40 & 50 \end{bmatrix} \Rightarrow \begin{bmatrix} 10 & 0 \\ 40 & 50 \end{bmatrix}$

```

int main(){
    int i, j, rows, columns, arr[10][10];
    cout << "Enter number of rows and columns" << endl;
    cin >> rows >> columns;
    cout << "Enter elements" << endl;
    for (rows = 0; rows < 1; rows++) {
        cout << endl;
        for (columns = 0; columns < j; columns++) {
            cout << arr[rows][columns];
            if (rows >= columns) {
                cout << arr[rows][columns];
            } else {
                cout << "0";
            }
        }
    }
}

```

6). Scalar multiplication

```

for (i = 0; i < rows; i++) {
    for (j = 0; j < columns; j++) {
        mul[i][j] = n * mul[i][j];
    }
}

```

Scalar Multiplication

7). Sum of Diagonal Elements.

```
int main() {
```

```
    int i, j, row, columns, a[10][10], sum = 0;
```

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

```
        sum = sum + a[i][i];
```

```
}
```

8). Sum of Each row & column of a matrix.

```
int main() {
```

```
    int i, j, row, columns, a[10][10], sum;
```

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

```
        sum = 0;
```

```
        for (j = 0; j < columns; j++) {
```

```
            sum = sum + a[i][j];
```

```
}
```

```
        cout << "Sum of elements of row " << i << sum;
```

```
    }
```

```
    for (i = 0; i < columns; i++) {
```

```
        sum = 0;
```

```
        for (j = 0; j < row; j++) {
```

```
            sum = sum + a[j][i];
```

```
}
```

```
        cout << "Sum of elements of column " << i << sum;
```

```
}
```

Q). Check whether a matrix is symmetric or not.

```
int main() {
    int i, j, rows, columns, a[10][10], b[10][10],
        count = 1;
```

```
    for (i = 0; i < rows; i++) {
        for (j = 0; j < columns; j++) {
            b[j][i] = a[i][j];
        }
    }
```

```
    for (i = 0; i < rows; i++) {
        for (j = 0; j < columns; j++) {
            if (a[i][j] != b[i][j]) {
                count++;
                break;
            }
        }
    }
```

```
    if (count == 0)
```

```
        cout << "Symmetric Matrix"
    else
```

```
        cout << "Not a Symmetric Matrix"
}
```

10) Transpose of a Matrix.

~~and void mainly~~

```
int i,j,rows,columns,a[10][10],b[10][10];
```

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

```
for(j=0;j<columns;j++)
    b[i][j] = a[i][c[j]]
```

1. What is the capital of France?

2

print b[i:j:j] //

1) Multiplication of a matrix

int main() {

int a[10][10], b[10][10], mul[10][10], r, c, i, j, k;
Read the matrix:

Read two matrices;

```
for (i=0; i<r; i++)
```

~~for(j=0; j < r; j++) {~~

$$m \ddot{u} [i,j;c;j] = 0;$$

for (k=0 ; k < c ; k++) {

$$3 \quad \text{matrix} f = a(i,j,k) + b(k,j,i)$$

3

Searching & Sorting Algorithms:

S	M	T	W	T	F	S	SONA HOOPA
Page No.:		Date:	11				

1). Linear Search.

```
int main() {
    int a[20], i, x, n;
    cout << "How many elements? ";
    cin >> n;
    cout << "Enter array elements: ";
    for (i=0; i<n; i++) {
        cin >> a[i];
    }
    cout << "Enter element to search: ";
    cin >> x;
    for (i=0; i<n; i++) {
        if (a[i] == x) {
            cout << "Element found at index " << i;
            return 0;
        }
    }
    cout << "Element not found";
    return 0;
}
```

Algorithm:

Q). Binary Search.

```

int main(){
    int i, low, high, mid, n, key, a[10];
    cout << "Enter no. of elements";
    cin >> n;
    cout << "Enter " << n << " integers";
    for (i=0; i<n; i++){
        cin >> a[i];
    }
    cout << "Enter value to find";
    cin >> key;

    low = 0;
    high = n - 1;
    mid = (low + high) / 2;

    while (low <= high){
        if (a[mid] == key)
            low = mid + 1;
        else if (a[mid] == key)
            cout << key << " found at location " << mid
            3 break;
        else{
            high = mid - 1;
            mid = (low + high) / 2;
        }
    }

    if (low > high)
        cout << " Not Found ";
    3
    return 0;
}

```

Time and Space complexity Analysis of Linear and Binary search.

Searching Algorithm.	Best case	Worst case	Average case	Space Complexity.
----------------------	-----------	------------	--------------	-------------------

1) Linear search $O(1)$. $O(n)$. ~~$O(n)$~~ $O(n)$.

2) Binary Search $O(1)$. $O(\log n)$. $O(\log n)$. $O(1)$.

1). Bubble sort.

```
int main(){
    int a[50], num, i, j, temp;
    cout << "How many elements?";
    cin >> num;
    cout << "Enter " << num << " elements";
    for (i=0; i<n; i++)
        cin >> a[i];
```

```
for (i=0; i<num-1; i++) {
    for (j=0; j<num-i-1; j++) {
        if (a[j] > a[j+1]) {
            temp = a[j];
            a[j] = a[j+1];
            a[j+1] = temp;
        }
    }
}
```

```
cout << "Array elements after Bubble sort are";
for (i=0; i<num; i++)
    cout << a[i];
cout << endl;
return 0;
}.
```

2). Selection sort.

```

int main() {
    int n;
    cin >> n;
    int arr[n];
    int i, j, position, swap;
    for (i = 0; i < n; i++) {
        cin >> arr[i];
    }
    for (i = 0; i < n - 1; i++) {
        position = i;
        for (j = i + 1; j < n; j++) {
            if (arr[position] > arr[j])
                position = j;
        }
        cout << "After 1st iteration " << endl;
        if (position != i) {
            swap = arr[i];
            arr[i] = arr[position];
            arr[position] = swap;
        }
    }
    cout << endl;
    for (i = 0; i < n; i++) {
        cout << arr[i] << " ";
    }
    return 0;
}

```

3). Insertion sort.

```
int main() {
    int a[100], number, i, j, temp;
    cout << "How many elements? ";
    cin >> number;
    cout << "Enter elements ";
    for (i = 0; i < number; i++)
        cin >> a[i];
}
```

```
for (i = 1; i <= number - 1; i++) {
    for (j = i; j > 0 & a[j - 1] > a[j]; j--)
```

```
    temp = a[j];
    a[j] = a[j - 1];
    a[j - 1] = temp;
}
```

```
}
```

```
cout << " Insertion Sort Result: ";
for (i = 0; i < number; i++) {
    cout << " " << a[i];
}
```

```
} //End of main()
return 0;
```

3-

4). Merge Sort.

```

int main()
{
    int arr[30], temp[30], i, j, k, n, size, l1, h1, l2, h2;
    cin >> n;
    for (i = 0; i < n; i++)
        cout << arr[i];
    cout << endl;

    for (size = 1; size < n; size = size * 2) {
        l1 = 0;
        k = 0;
        while (l1 + size <= n) {
            h1 = l1 + size - 1;
            l2 = h1 + 1;
            h2 = l2 + size - 1;
            if (h2 >= n)
                h2 = n - 1;
            i = l1;
            j = l2;
            while (i <= h1 && j <= h2) {
                if (arr[i] <= arr[j])
                    temp[k++] = arr[i++];
                else
                    temp[k++] = arr[j++];
            }
            while (i <= h1)
                temp[k++] = arr[i++];
            while (j <= h2)
                temp[k++] = arr[j++];
        }
        l1 = h2 + 1; // Take next 2 pairs for merging
    }
}

```

3.

```
// Many pairs left  
for (i = j + 1; k < n; i++)  
    if (arr[i] > arr[j])
```

```
for (i=0; i<n; i++)  
    arr[i] = temp[i];
```

```
cout << "sorted list is ";
```

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

return 0;

Quick Sort:

```
#include <stdio.h>
```

```
void quicksort(int a[], int first, int last) {
    int i, j, pivot, temp;
    if (first < last) {
        pivot = first;
```

```
        i = first; j = last;
```

```
        while (i < j) {
```

```
            if (a[i] <= a[pivot] && i < last)
                i++;
```

```
            while (a[j] > a[pivot])
                j--;
```

```
            if (i < j) {
```

```
                temp = a[i];
                a[i] = a[j];
```

```
                a[j] = temp;
            }
```

```
}
```

```
3
```

```
temp = a[pivot];
```

```
a[pivot] = a[j];
a[j] = temp;
```

```
quicksort(a, first, j-1);
quicksort(a, j+1, last);
```

```
3
```

```
3
```

```

int main(){
    int i, count, a[25];
    cout << "How many elements?";
    cin >> count;
    cout << "Entered " << count << " elements";
    for (i=0; i<count; i++)
        cin >> a[i];
    quicksort(a, 0, count - 1);
    cout << "Order of sorted Elements:";
    for (i=0; i<count; i++)
        cout << a[i];
    cout << endl;
    return 0;
}

```

3.

Time and space complexities of sorting Algorithms:

Sorting : Best Worst Average Space
Algorithm case. case case Complexity.

1) Bubble sort n n^2 n^2 1.

2) Selection sort n^2 n^2 n^2 1.

3) Insertion sort n n^2 n^2 1.

4) Merge sort $n \log n$ $n \log n$ $n \log n$ n.

5) Quick sort $n \log n$ n^2 $n \log n$. $\log n$.

—X—