

ASSIGNMENT 2:

NAME: Vanshaj Goel ROLL NUMBER: 1024030366

QUE1: Implement the Binary search algorithm regarded as a fast search algorithm with run-time complexity of $O(\log n)$ in comparison to the Linear Search.

CODE:

```
1  #include<bits/stdc++.h>
2  using namespace std;
3  void display (int arr[] , int size){
4      for (int i = 0; i<size ; i++){
5          cout << arr[i]<<" ";          //displaying the array //
6      }
7      cout<<endl;
8  }
9  void binarysearch(int arr[], int size , int target){    // implementing the binary search //
10     int s =0;
11     int e = size -1 ;
12     while(s<=e){
13         int mid = s + (e-s)/2;
14         if (arr[mid]== target){
15             cout <<"the element is at index "<<mid <<endl;
16             return;
17         }else
18         if(arr[mid]<target){
19             s = mid +1 ;
20         }
21         else{
22             e = mid -1 ;
23         }
24     }
25     cout <<"element not found in array "<<endl;
26 }
27
28
29 int main (){
30     cout <<"the array is "<<endl;
31     int arr[7] = { 10, 11, 15 , 16 , 18 , 20 , 30};
32     display ( arr , 7);
33     cout <<"enter the target element "<<endl;
34     int target ;
35     cin >> target ;
36     binarysearch( arr , 7 , target );
37     return 0 ;
38 }
```

```
the array is
10 11 15 16 18 20 30
enter the target element
16
the element is at index 3
```

-----X-----X-----X-----

QUE2: Bubble Sort is the simplest sorting algorithm that works by repeatedly swapping the adjacent elements if they are in wrong order. Code the Bubble sort with the following elements:
64 34 25 12 22 11 90

CODE:

```
1  #include<bits/stdc++.h>
2  using namespace std;
3  void display (int arr[] , int size){
4      for (int i = 0; i<size ; i++){
5          cout << arr[i]<<" ";           //displaying the array //
6      }
7      cout<<endl;
8  }
9  void bubblesort( int arr[], int size ){
10     for (int i =0; i<size ; i++){
11         for (int j =0 ; j<size -1 -i ; j++){
12             if(arr[j]>arr[j+1]){
13                 swap(arr[j], arr[j+1]);
14             }
15         }
16     }
17 }
18
19 int main(){
20     int arr[7]={64 , 34 , 25 , 12 , 22 , 11 , 90};
21     cout <<"current array is "<<endl;
22     display ( arr , 7);
23     cout <<"after applying bubble sort "<<endl;
24     bubblesort(arr , 7 );
25     display ( arr , 7);
26
27     return 0;
28 }
```

```
current array is
64 34 25 12 22 11 90
after applying bubble sort
11 12 22 25 34 64 90
```

-----X-----X-----X-----

QUE3: Given an array of n-1 distinct integers in the range of 1 to n, find the missing number in it in a Sorted Array

(a) Linear time

(b) Using binary search.

CODE:

```

1  #include<bits/stdc++.h>
2  using namespace std;
3  void display (int arr[] , int size){
4      for (int i = 0; i<size ; i++){
5          cout << arr[i]<<" "; //displaying the array //
6      }
7      cout<<endl;
8  }
9
10 int missing(int arr[], int size){ //linear search //
11     int sum =0;
12     for(int i =0; i<size;i++){
13         sum+=arr[i];
14     }
15     return sum ;
16 }
17
18 int binary(int arr[], int size ){ // implementing the binary search //
19     int s =0;
20     int e = size -1 ;
21     int ans =0;
22     while(s<=e){
23         int mid = s + (e-s)/2;
24         if(arr[mid]==mid+1){
25             s= mid +1;
26         }else{
27             ans=mid;
28             e = mid -1;
29         }
30     }
31     return ans+1 ;
32 }
33
34 }
35

```

```

int main(){
    cout <<"the array is " <<endl;
    int arr[7]={ 1, 2 , 3 ,4 , 6 , 7 , 8}; //sorted array //
    display(arr , 7);
    int answer = missing(arr , 7);
    int x = 8; // value of n //
    int sum = x*(x+1)/2; // sum of first n natural number//
    int final = sum - answer ;
    cout<<"the missing number in the array from range 1 to 8 by linear search is " <<final <<endl;
    int binans= binary(arr , 7);
    cout<<"the missing number in the array from range 1 to 8 by binary search is " <<binans <<endl;
    return 0;
}

```

the array is

1 2 3 4 6 7 8

the missing number in the array from range 1 to 8 by linear search is 5

the missing number in the array from range 1 to 8 by binary search is 5

-----X-----X-----X-----

QUE4: String Related Programs

- (a) Write a program to concatenate one string to another string.
- (b) Write a program to reverse a string.
- (c) Write a program to delete all the vowels from the string.
- (d) Write a program to sort the strings in alphabetical order.
- (e) Write a program to convert a character from uppercase to lowercase.

CODE (a) :

```
1  #include<bits/stdc++.h>
2  using namespace std;
3  int main() {
4      string str1,str2;
5      cout<<"Enter first and second strings to be concatenated: "<<endl;
6      getline(cin,str1);
7      getline(cin,str2);
8      char res[100];
9      int i=0,j=0;
10     while(str1[i]!='\0'){
11         res[i]=str1[i];
12         i++;
13     }
14     while(str2[j]!='\0'){
15         res[i]=str2[j];
16         i++,j++;
17     }
18     res[i]='\0';
19     cout<<"Concatenated result is: "<<res;
20     return 0;
21 }
22
```

```
Enter first and second strings to be concatenated:
Vanshaj Goel
Vanshaj
Concatenated result is: Vanshaj Goel Vanshaj
```

CODE (b) :

```
1  #include<bits/stdc++.h>
2  using namespace std;
3
4  int main() {
5
6      string str;
7      cout<<"Enter the string to be reversed"<<endl;
8      getline(cin,str);
9
10     int i=0,j=str.length()-1;
11     while(j>i){
12         int temp=str[i];
13         str[i]=str[j];
14         str[j]=temp;
15
16         i++,j--;
17     }
18     cout<<"The reversed string is: "<<endl;
19     cout<<str;
20
21     return 0;
22 }
23
```

```

Enter the string to be reversed
Hello World
The reversed string is:
Dlrow olleH

```

CODE (c) :

```

1  #include<bits/stdc++.h>
2  using namespace std;
3  int main() {
4      string str;
5      cout<<"Enter the string whose vowels need to be removed"<<endl;
6      getline(cin,str);
7      int size=str.length();
8      for(int i=0;i<size;i++){
9          if(str[i]=='a' || str[i]=='e' || str[i]=='i' || str[i]=='o' || str[i]=='u'){
10             for(int j=i;j<size-1;j++){
11                 str[j]=str[j+1];
12             }
13             size--;
14         }
15     }
16     cout<<"The string after removing the vowels is: "<<endl;
17     for(int i=0;i<size;i++){
18         cout<<str[i];
19     }
20
21     return 0;
22 }
23

```

```

Enter the string whose vowels need to be removed
Hello aeiou
The string after removing the vowels is:
Hll eo

```

CODE (d) :

```

1  #include<bits/stdc++.h>
2  using namespace std;
3  int main() {
4      int arr[26]={};
5      string str;
6      cout<<"Enter the string you want to make in alphabetical order"<<endl;
7      getline(cin,str);
8      for(int i=0;i<str.length();i++){
9          arr[str[i]-'a']++;
10     }
11     cout<<"Final result: "<<endl;
12     for(int i=0;i<26;i++){
13         if(arr[i]>0){
14             while(arr[i]){
15                 char c=i+'a';
16                 cout<<c;
17                 arr[i]--;
18             }
19         }
20     }
21
22     return 0;
23 }
24
25

```

```
Enter the string you want to make in alphabetical order
Vanshaj
Final result:
aahjns
```

CODE (e) :

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      char ch;
6      cout << "Enter a character: ";
7      cin >> ch;
8
9      if (ch >= 'A' && ch <= 'Z') {
10         ch = ch + 32;
11     }
12
13     cout << "Lowercase: " << ch << endl;
14
15     return 0;
16 }
```

OUTPUT (e) :

```
Enter a character: A
Lowercase: a
```

5) Space required to store any two-dimensional array is *number of rows* × *number of columns*. Assuming an array is used to store elements of the following matrices, implement an efficient way that reduces the space requirement.

- (a) Diagonal Matrix.
- (b) Tri-diagonal Matrix.
- (c) Lower triangular Matrix.
- (d) Upper triangular Matrix.
- e) Symmetric Matrix

CODE A :

```
#include<iostream>
using namespace std;
int main(){
    int r,c;
    cout<<"Enter no. of rows:";
    cin>>r;
    cout<<"Enter no. of col: ";
    cin>>c;
    int arr[r];
    cout<<"Enter Diagonal Elements of matrix: ";
    for(int i=0;i<r;i++){
        cin>>arr[i];
    }
    cout<<"Your resultant matrix is: "<<endl;
    for(int i=0;i<r;i++){
        for(int j=0;j<c;j++){
            if(i==j){
                cout<<arr[i]<<" ";
            }else cout<<"0"<<" ";
        }cout<<endl;
    }

    return 0;
}
```

```
Enter no. of rows:2
Enter no. of col: 2
Enter Diagonal Elements of matrix: 1
2
Your resultant matrix is:
1 0
0 2
```

CODE B :

```
#include<iostream>
using namespace std;
int main(){
    int r,c;
    cout<<"Enter rows: ";
    cin>>r;
    cout<<"Enter col:" ;
    cin>>c;
    int s=3*r-2;
    int arr[s];
    cout<<"Enter elements: "<<endl;
    for(int i=0;i<s;i++){
        cin>>arr[i];
    }
    int k=0;
    cout<<"Resultant matrix is: "<<endl;
    for(int i=0;i<r;i++){
        for(int j=0;j<c;j++){
            if(i-j== -1 || i==j || i-j==1){
                cout<<arr[k]<<" ";k++;
            }else cout<<"0"<<" ";
        }cout<<endl;
    }

    return 0;
}
```

```
Enter rows: 3
Enter col:3
Enter elements:
1 2 3 4 5 6 7
Resultant matrix is:
1 2 0
3 4 5
0 6 7
```


CODE C :

```
#include<iostream>
using namespace std;
int main(){
    int r, c;
    cout << "Enter rows: ";
    cin >> r;
    cout << "Enter cols: ";
    cin >> c;

    int n = r;
    int s = (n * (n + 1)) / 2;
    int arr[s];

    cout << "Enter " << s << " elements (lower triangular matrix elements row-wise): " << endl;
    for (int i = 0; i < s; i++) {
        cin >> arr[i];
    }
    int k = 0;
    cout << "Resultant matrix is: " << endl;
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            if (i >= j) {
                cout << arr[k] << " ";
                k++;
            } else {
                cout << "0 ";
            }
        }
        cout << endl;
    }
    return 0;
}
```

```
Enter rows: 3
Enter cols: 3
Enter 6 elements (lower triangular matrix elements row-wise):
1 2 3 4 5 6
Resultant matrix is:
1 0 0
2 3 0
4 5 6
```

CODE 4 :

```
#include<iostream>
using namespace std;
int main(){
    int r, c;
    cout << "Enter rows: ";
    cin >> r;
    cout << "Enter cols: ";
    cin >> c;

    int n = r;
    int s = (n * (n + 1)) / 2;
    int arr[s];

    cout << "Enter " << s << " elements (upper triangular matrix elements row-wise): " << endl;
    for (int i = 0; i < s; i++) {
        cin >> arr[i];
    }
    int k = 0;
    cout << "Resultant matrix is: " << endl;
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            if (i <= j) {
                cout << arr[k] << " ";
                k++;
            } else {
                cout << "0 ";
            }
        }
        cout << endl;
    }
    return 0;
}
```

```
Enter rows: 3
Enter cols: 3
Enter 6 elements (upper triangular matrix elements row-wise):
1 2 3 4 5 6
Resultant matrix is:
1 2 3
0 4 5
0 0 6
```

CODE 5 :

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

int main() {
    int r, c;
    cout << "Enter rows: ";
    cin >> r;
    cout << "Enter cols: ";
    cin >> c;
    if (r != c) {
        cout << "Symmetric matrix must be square!" << endl;
        return 0;
    }
    int n = r;
    int s = (n * (n + 1)) / 2; // only lower triangle needed
    int arr[s];
    cout << "Enter " << s << " elements (lower triangular part row-wise): " << endl;
    for (int i = 0; i < s; i++) {
        cin >> arr[i];
    }
    cout << "Resultant Symmetric Matrix is: " << endl;
    int k = 0; // index in arr[]
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            if (i >= j) {
                cout << arr[k] << " "; // lower triangle
                k++;
            } else {
                int index = (j * (j + 1)) / 2 + i;
                cout << arr[index] << " ";
            }
        }
        cout << endl;
    }
    return 0;
}
```

```
Enter rows: 3
Enter cols: 3
Enter 6 elements (lower triangular part row-wise):
2 3 4 5 6 7
Resultant Symmetric Matrix is:
2 3 5
3 4 6
5 6 7
```

-----X-----X-----X-----

QUE7: Let $A[1 \dots n]$ be an array of n real numbers. A pair $(A[i], A[j])$ is said to be an inversion if these numbers are out of order, i.e., $i < j$ but $A[i] > A[j]$. Write a program to count the number of inversions in an array.

CODE:

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

```
#include<iostream>
using namespace std;
int main(){
    int arr[9]={1,8,9,1,4,0,18,34,20};
    int count=0;
    for(int i=0;i<9;i++){
        for(int j=i+1;j<9;j++){
            if(arr[i]>arr[j]){
                count++;
            }
        }
    }
    cout<<"No. of inversion: "<<count<<endl;

    return 0;
}
```

No. of Inversion in this case is 10.

-----X-----X-----X-----

QUE8: Write a program to count the total number of distinct elements in an array of length n.

CODE:

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

int main() {
    int n;
    cout << "Enter the number of elements: ";
    cin >> n;

    int arr[n];
    cout << "Enter " << n << " elements:\n";

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

    set<int> distinctElements;

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

    cout << "Total number of distinct elements: " << distinctElements.size() << endl;

    return 0;
}
```

```
Enter the number of elements: 8
Enter 8 elements:
1 1 2 3 4 4 4 5
Total number of distinct elements: 5
```

-----X-----X-----X-----