## **ASSIGNMENT 2:**

NAME: Ujjwal Pant

ROLL NUMBER: 1024030370



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.

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

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

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

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

```
#include<bits/stdc++.h>
             using namespace std;
   2
   3
                          void display (int arr[] , int size){
                          for (int i = 0; i<size ; i++){
   4
                                     cout << arr[i]<<" ";
   6
                          cout<<endl;
   8
             int missing(int arr[], int size){
10
11
                          int sum =0;
                          for(int i =0; i<size;i++){</pre>
12
                                     sum+=arr[i];
13
14
                          return sum ;
16
17
             int binary(int arr[], int size ){
19
                          int s = 0;
                          int e = size -1;
20
                          int ans =0;
21
22
                          while(s<=e){
23
                                      int mid = s + (e-s)/2;
24
                                       if(arr[mid]==mid+1){
25
                                                              s= mid +1;
26
27
                                                  ans=mid;
                                                   e = mid -1;
28
29
30
31
32
                          return ans+1;
 33
 34
35
             int main(){
36
37
                          cout <<"the array is "<<endl;</pre>
38
                          int arr[7]={ 1, 2 , 3 ,4 , 6 , 7 , 8};
39
                          display(arr , 7);
                          int answer = missing(arr , 7);
40
41
                          int x = 8;
42
                          int sum = x*(x+1)/2;
43
                          int final = sum - answer ;
                          cout<<"the missing number in the array from range 1 to 8 by linear search is " <<final <<endl;</pre>
44
45
                          int binans= binary(arr , 7);
                          cout<<"the missing number in the array from range 1 to 8 by binary search is " <<bir>binary search is " <<br/>is " <br/>is " <<br/>is " <br/>is " <b
46
47
                          return 0;
```

## 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.

a)

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

b)

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

c)

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

d)

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

e)

```
#include <iostream>
1
    using namespace std;
2
3
    int main() {
4
5
         char ch;
         cout << "Enter a character: ";</pre>
6
7
         cin >> ch;
8
9
         if (ch >= 'A' && ch <= 'Z') {
10
             ch = ch + 32;
11
12
         cout << "Lowercase: " << ch << endl;</pre>
13
14
15
         return 0;
16
17
```

- 5) Space required to store any two-dimensional array is  $number\ of\ rows \times number\ of\ columns$ . 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

a)

```
#include<iostream>
   using namespace std;
3
    int main(){
        int r,c;
4
        cout<<"Enter no. of rows:";</pre>
5
6
        cin>>r;
        cout<<"Enter no. of col: ";</pre>
        cin>>c;
8
        int arr[r];
9
         cout<<"Enter Diagonal Elements of matrix: ";</pre>
10
         for(int i=0;i<r;i++){
11
12
             cin>>arr[i];
13
         cout<<"Your resultant matrix is: "<<endl;</pre>
14
15
         for(int i=0;i<r;i++){
16
             for(int j=0;j<c;j++){</pre>
                  if(i==j){
17
                      cout<<arr[i]<<" ";
18
                  }else cout<<"0"<<" ";
19
20
             }cout<<endl;</pre>
21
22
23
        return 0;
24
```

b)

```
#include<iostream>
   using namespace std;
    int main(){
3
4
         int r,c;
         cout<<"Enter rows: ";</pre>
 5
 6
         cin>>r;
         cout<<"Enter col:";</pre>
8
         cin>>c;
9
         int s=3*r-2;
10
         int arr[s];
11
         cout<<"Enter elements: "<<endl;</pre>
12
         for(int i=0;i<s;i++){
13
             cin>>arr[i];
14
         int k=0;
15
16
         cout<<"Resultant matrix is: "<<endl;</pre>
17
         for(int i=0;i<r;i++){
18
             for(int j=0;j<c;j++){</pre>
19
                  if(i-j==-1 || i==j || i-j==1){
                      cout<<arr[k]<<" ";k++;
20
21
                  }else cout<<"0"<<" ";</pre>
22
             }cout<<endl;</pre>
23
24
25
26
         return 0;
27
```

c)

```
#include<iostream>
     using namespace std;
     int main(){
         int r, c;
cout << "Enter rows: ";</pre>
          cin >> r;
cout << "Enter cols: ";</pre>
8
          cin >> c;
9
10
          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++) {
14
15
16
              cin >> arr[i];
          int k = 0;
18
          cout << "Resultant matrix is: " << endl;
for (int i = 0; i < n; i++) {</pre>
19
20
               for (int j = 0; j < n; j++) {
    if (i >= j) {
23
                         cout << arr[k] << " ";
24
                         k++;
                      else {
26
                         cout << "0 ";
27
28
29
               cout << endl;</pre>
30
31
          return 0;
32
```

d)

```
#include<iostream>
     using namespace std;
     int main(){
         int r, c;
cout << "Enter rows: ";</pre>
         cin >> r;
cout << "Enter cols: ";</pre>
 6
         cin >> c;
8
 9
10
         int n = r;
int s = (n * (n + 1)) / 2;
12
          int arr[s];
13
         cout << "Enter " << s << " elements (upper triangular matrix elements row-wise): " << endl; for (int i = 0; i < s; i++) {
14
15
16
            cin >> arr[i];
17
18
          int k = 0;
          cout << "Resultant matrix is: " << endl;</pre>
19
          for (int i = 0; i < n; i++) {
20
              for (int j = 0; j < n; j++) {
    if (i <= j) {
21
23
                        cout << arr[k] << " ";
24
                        k++;
                     else {
26
                        cout << "0 ";
27
28
29
               cout << endl;</pre>
30
32
```

e)

```
#include<iostream>
    using namespace std;
    int main() {
 5
         int r, c;
 6
         cout << "Enter rows: ";</pre>
         cin \gg r;
         cout << "Enter cols: ";</pre>
 8
 9
         cin >> c;
10
         if (r != c) {
             cout << "Symmetric matrix must be square!" << endl;</pre>
11
12
             return 0;
13
14
         int n = r;
15
         int s = (n * (n + 1)) / 2; // only lower triangle needed
16
         int arr[s];
17
         cout << "Enter " << s << " elements (lower triangular part row-wise): " << endl;</pre>
         for (int i = 0; i < s; i++) {
18
19
             cin >> arr[i];
20
21
         cout << "Resultant Symmetric Matrix is: " << endl;</pre>
22
         int k = 0; // index in arr[]
23
         for (int i = 0; i < n; i++) {
24
             for (int j = 0; j < n; j++) {
25
                 if (i >= j) {
26
                      cout << arr[k] << " "; // lower triangle</pre>
27
                      k++;
28
                 } else {
29
30
                      int index = (j * (j + 1)) / 2 + i;
31
                      cout << arr[index] << " ";</pre>
32
33
34
             cout << endl;</pre>
35
36
37
         return 0;
38
```

QUE7: Let A[1 .... 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

```
#include<iostream>
    using namespace std;
  vint main(){
 4
        int arr[9]={1,8,9,1,4,0,18,34,20};
 5
         int count=0;
 6
         for(int i=0;i<9;i++){
 7
             for(int j=i+1;j<9;j++){
 8
                 if(arr[i]>arr[j]){
 9
                     count++;
10
11
12
         cout<<"No. of inversion: "<<count<<endl;</pre>
13
14
15
        return 0;
16
```

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

```
#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;
}</pre>
```