LAB 3

Q1 WAP to insert new element at given index number in the array.

```
/*************************************************************************
//This program is developed by Aman Singh Rawat (221B056)
/*************************************************************************
#include <iostream>
using namespace std;

void add_ele(int arr[], int n, int ele, int index){
        int new_arr[n+1];
        n+=1;
        for(int i =0;i< 10; i++){
                new_arr[i] = arr[i];
        }
        for(int i = n-1; i>index; i--){
                new_arr[i] = new_arr[i-1];
        }
        new_arr[index] == ele;
        for(int i = 0 ; i<n; i++) cout <<" " << new_arr[i];
}

int main(){
        int n = 10;
        int arr[] = {1,4,3,2,4,3,4,1,2,8};
        cout << "Enter the element and the index: ";
        int ele, index;
        cin >> ele >> index;
        cout << "The array is: ";
        for(int i = 0; i<10; i++){
                cout << arr[i] << " ";
        }
        add_ele(arr, n , ele, index);
}
```

Q2 WAP to implement the linear search. Use function concept, if element is found then return index number of element otherwise return -1;

```
/*************************************************************************
//This program is developed by Aman Singh Rawat (221B056)
/*************************************************************************
#include <iostream>
using namespace std;

int search_ele(int arr[], int n, int ele){
        for(int i = 0; i<n; i++){
                if(arr[i] == ele)
                        return i;
        }
        return -1;
}

int main(){
        int n = 10;
        int arr[] = {1,4,3,2,4,3,4,1,2,8};
        cout << "Enter the element to search: ";
        int ele;
        cin >> ele;
        cout << "Returned index value is: ";
        cout<<search_ele(arr, n , ele) << endl;

}
```

Q3 WAP to delete an element from an array, use search algorithm to find the index number of given number; if element to be deleted is not found then print "Error: element not found".

```
/*************************************************************************
//This program is developed by Aman Singh Rawat (221B056)
/*************************************************************************
#include <iostream>
using namespace std;

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

```
        cout <<"Error: element not found";
        return -1;
}

void del_ele(int arr[], int n, int index){
                for(int i = index; i<n-1; i++){
                        arr[i] = arr[i+1];
        }

        for(int i = 0 ; i<n-1; i++){
                cout <<" " << arr[i];
        }
}

int main(){
        int n = 10;
        int arr[] = {1,4,3,2,4,3,4,1,2,8};
        cout << "Enter the element to search: ";
        int ele;
        cin >> ele;
        cout << "Returned index value is: ";
        int index = search_ele(arr,n,ele);
        cout << index << endl;
        if(index == -1){
                return 0;
        }
        else{
                del_ele(arr, n,index);
        }

}
```

Q4 WAP for checking whether there are any duplicated elements in the array or not?

```
/*********************************************************************
//This program is developed by Aman Singh Rawat (221B056)
/*********************************************************************
#include <iostream>
using namespace std;

int check_dup(int arr[], int n){
        for(int i = 0; i<n; i++){
```

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

        return 0;
}



int main(){
        int arr[] = {12,423,423,11,89,23};
        int n = 6;
        if(check_dup(arr, n) == 1){
            cout << "Duplicates found!" << endl;
        }
        else cout << "No Duplicates found!";
}
```

Q5 Mary is a kindergarten teacher. She has given a task to the children after teaching them a list of words. The task is to find the unknown words (other than the words they already know) from the given text. Write a function which accepts the text and the known list of words and prints a set of unknown words found. If there are no unknown words found then prints "Successful". [Hint use find_word() function of Lab 1]

Sample Input
Text: "the sun rises in the east"
Vocabulary: ["sun","in","east","doctor","day"]
Expected Output
{"rises", "the"}

```
/*****************************************************************
//This program is developed by Aman Singh Rawat (221B056)
/*****************************************************************
#include <iostream>
using namespace std;

int find(string sen, string word){
        int tmp;
```

```
        char ch= ' ';
        for(int i =0; i<sen.length(); i++){
                if((sen[i] == ch || sen[i+word.length()] == ch) || sen[i+word.length()] == '\0'){
                        tmp =1;
                        for(int j=0;i<word.length(); j++){
                                if(sen[i+j+1] == word[j]){
                                        tmp = 0;
                                        break;
                                }
                        }
                        if (tmp == 1){
                                return 1;
                        }
                }
        }
        return 0;
}

string* find_unknowns(string sen, string arr[],int n){
        static string u_arr[100];
        int counter = 0;
        for(int i = 0; i<n; i++){
                if(find(sen, arr[i]) == 0){
                        u_arr[counter++] = arr[i];
                }
        }
        return u_arr;
}

int main(){
        string srr[100];
        cout << "Enter the known words, enter 'end' to stop entering words: ";
        int a = 0;
        string tmp;
        cin >> tmp;
        do{
                srr[a++] = tmp;
                cin >> tmp;
        }
        while (tmp != "end");
```

```
        cout << "Enter the string: ";
        string sen;
        cin >> sen;
        string* u_words;
        u_words = find_unknowns(sen,srr,a);
        cout << "Unknowns are: ";
        for(int i = 0; i<100; i++){
                cout<< u_words[i] << " ";
        }
        cout << endl;
}
```