INDIRA GANDHI NATIONAL OPEN UNIVERSITY

इन्दिरा गाँधी राष्ट्रीय मुक्त विश्वविद्यालय

ENROLLMENT NO. : 2350623568

NAME : Ritk Kumar

ADDRESS : H. NO. 121 Meethapur village, Jaitpur mod,
Badarpur New Delhi - 110044

PROGRAMME TITLE : Bachler of Computer Application

COURSE TITLE : Algorithm Design Lab

COURSE CODE : BCSL-045

REGIONAL CENTRE CODE : 07

STUDY CENTRE & CODE : Mohyal Education and Research
Institute of Tech(MERIT), Code - 07162P

PHONE NO. : 8287385862

EMAIL ID : neerajaccout@gmail.com

DATE : 26/04/2025 SIGNATURE : RITIK KUMAR

ignou
THE PEOPLE'S UNIVERSITY

इंदिरा गांधी राष्ट्रीय मुक्त विश्वविद्यालय
मैदान गढ़ी, नई दिल्ली - 110068
Indira Gandhi National Open University
Maidan Garhi,New Delhi - 110068

IGNOU - Student Identity Card

Enrolment Number :	**2350623568**
RC Code :	**07: DELHI 1 (MOHAN ESTATE (SOUTH DELHI))**
Name of the Programme :	**BCA : BACHELOR OF COMPUTER APPLICATIONS**
Name :	**RITIK KUMAR**
Guardian's Name :	**JUJHAR SINGH**
Address :	**House no 121 Meethapur Village, Jaitpur Mod Badarp DELHI**
Pin Code :	**110044**

Instructions :
1. This card should be produced on demand at the Study Center, Examination Center or any other Establishment of IGNOU to use its facilities.
2. The facilities would be available only relating to the Programme/course for which the student is registered.
3. This ID Card is generated online. Students are advised to take a color print of this ID Card and get it laminated.
4. The student details can be cross checked with the QR Code at www.ignou.ac.in. PD 2024-12-08

Registrar
Student Registration Division

Course Code	:	**BCSL-045**
Course Title	:	**Introduction to Algorithm design Lab**
Assignment Number	:	**BCA(IV)/L-045/Assignment/2024-25**
Maximum Marks	:	**50**
Weightage	:	**25%**
Last date of Submission	:	**31st October, 2024 (For July Session)**
	:	**30th April, 2025 (For January Session)**

Note: Answer all the questions which carry 40 marks. All questions are of equal marks.The rest 10 marks are for viva voce. You are required to write programs in C-language for all the problems , execute and show the results. You may use illustrations and diagrams to enhance the explanations. Please go through the guidelines regarding assignments given in the Programme Guide for the format of presentation. Make suitable assumption if necessary.

Q1. Implement the Insertion Sort algorithm for sorting the following list of numbers in the ascending order, showing the list obtained at each step:

27, 15, 42, 3, 9, 29, 81, 54, 0, 13

Also calculate the total number of exchange operations and how many times the loop will execute in this algorithm. **(8 Marks)**

Q2. Write a C program to implement the binary search algorithm. The program should first sort an array using any sorting algorithm and then search for a given element. Also, understand its efficiency. **(8 Marks)**

Q3. Write a program to implement to reverse the following a given 5-digit number and calculate the total number of times the loop construct executed. **(8 Marks)**

Q4. Write a C program to implement ***a stack*** using a linked list with push, pop, and display operations. **(8 Marks)**

Q5. Write a C program to implement a binary tree and perform in-order, pre-order, and post-order traversals. Also, understand the efficiency of the program. **(8 Marks)**

Course code BCSL 045
Course Title: Introduction to Algorithm design Lab.

Que → 1 Implement the Insertion sort algorithm for following list of numbers in the ascending order, showing the list obtained at each step.

27, 15, 42, 3, 9, 29, 81, 54, 0, 13

Also calculate the total number of exchange operations and how many times the loop will execute in this algorithm.

Ans The C implementation of the Insertion Sort algorithm, including output that shows the state of the list at each step, and calculation of the total number of exchange operations and the number of loop executions.

program:

```
#include <stdio.h>
//function to perform Insertion Sort and
track the number of exchange and loop
Executions.
Void InsertionSort (int arr[], int n) {
    int i, j, key;
```

```
int exchangecount = 0; // To Track the number
                          of exchanges.
int loopcount = 0; // To track the number
                      of loop executions.

//perform Insertion Sort
    for (i = 1; i<n; i++){
        key = arr [i];
        j = i - 1;

// print array before the current insertion
    printf ("step %d: ", i);
    for (int k = 0; k<n; k++) {
        print ("%d", arr [k]);
    }
        print ("\n");

// Move Elements of arr [0..i-1] that
   are greater than key.
// to one position ahead of their current
        position.

while (j >= 0 && arr[j] > key) {
        arr [j+1] = arr [j];
        j--;
```

```
exchangeCount++; // count exchange operations.
loopcount++; // count loop executions in the
                Inner loop.
    arr[j+1] = key;
    loopcount++; // count the Final Inner
                loop execution (False conditions)
}
// print final sorted array.
print("Final Sorted array:");
for (i=0; i<n; i++) {
    print("%d", arr[i];
}
    print("\n");

// print statistics
Printf("Total number of exchanges: %d\n",
                exchangeCount);
Printf("Total number of loop executions: %d\n"
                loopcount);
}
int main() {
// Array of number to be sorted
int arr[] = {27, 15, 42, 3, 9, 24, 81, 54
                0, 13};
int n = sizeof(arr) / sizeof(arr[0]);
```

```
int n = sizeof(arr)/sizeof(arr[0]);

//perform insertion sort on the array
insertionSort(arr, n);

return 0;

}
```

Que → 2. write a C program to implement the binary search algorithm. The program should first sort an array using any sorting algorithm and then search for a given element. Also, understand its efficiency.

Ans A C program that implements the Binary search algorithm. The program first sort and then performs a binary to first given element. Additionally, the program evaluates the efficiency of both sorting and searching operations.

Program :

```
#include <stdio.h>
// function to perform insertions sort
                    on the array.
```

Page No. _____
Date

```
// Function to perform insertion Sort on the
                array.
void InsertionSort (int arr [], int n) {
    int , key , j ;
    for (i = 1; i <n ; i++) {
      key = arr [i]

    j = i - 1 ;

// Move Elements of arr (0...i-1 ] that
      greater than key to one position
                ahead.
while (j >= 0 && array (j ] > key ) {
      arr (j+1) = arr[j ];
        j = j - 1 ;

    }
arr [j+1 ] = key
   }
}

// function to perform binary Search on the
      Sorted array
int binary Search arr [], int right , int
                terget ) {
while (left <= righ ) {
```

```
while (left <= right) {
    int mid = left + (right - left) / 2

// Check if target is present at mid
    if (arr[mid] == target)
        return mid;

// If target is greater ignore the left half
    if (arr[mid] < target)
        left = mid + 1;

// If target is smaller, ignore the
    right half
    else
        right = mid - 1;
    }

// Target not found
        return -1;
    }
int main() {
    int n, target, result;

//Array to be Sorted and Searched.
```

Page No. _____
Date

```
Printf("Enter the number of elements in
  the array : ");

    Scanf("%d, &n);

  int arr[n];
  Print("Enter the number of Elements
    in the array : ");

  int arr[n];
  printf("Enter %d Element : \n", n);
     for(int i=0; i<n; i++) {
        scanf("%d", &arr[i]);
        ~~scanf("~~
     }

// Sorting the array using insertion sort
     insertionSort(arr, n);

// Display Sorted array.
  printf("Sorted array : ");
     for(int i=0; i<n; i++) {
        printf("%d", arr[i]);
     }
```

```
printf("\n");

// Input the element to be searched
printf("Enter the element for: ");
scanf("%d", &target);

// perform Binary search
result = binarySearch(arr, 0, n-1, target, result);
} else {
    printf("Element %d not found in the array\n",
                target);
    }

    return 0;
}
```

Ques→3. Write a program to implement to following a given 5-digit number and calculate the total number the loop construct executed.

Ans A C program that reverses a given 5-digit number and calculates the total number of times the loop construct executed.

Page No.
Date

Program :-

```
#include <stdio.h>

int main() {
int num, reverseNum = 0, remainder, loopcount = 0;

//Input a 5-digit number
printf("Enter a 5-digit number:");
    scanf("%d",&num)

//check if the input number is a 5-digit
number.
    if (num <10000 || num > 99999) {

    Printf("Please Enter a valid 5-digit number
                .\n");
    return 1; //Exit the program if the
                input is invalid

}
int originalNum = num; //Store the original
                number for display.

//Reverse the number using a loop.
```

Page No.
Date

```
while (num != 0) {
    remainder = num % 10;   // Get the last digit
                               of the number.
    reversedNum = reversedNum * 10 + remainder; // Build
                               the reversed Number
    num = num / 10;         // Remove the last digit from
                               the number.
    loopCount ++;           // Increment the loop counter
}

// Display the reversed Number.
printf("Reversed number : %d \n", reversedNum);

// Display the number of times the loop executed
printf("Total number of loop execution : %d \n"
                , loopCount);

    return 0;
}
```

Page No.
Date

Que→4 write a C program to implement a stack using a linked list with push, pop, and display operations.

Ans A C program to implement a stack using a linked list. The program includes the basic stack operations: push, pop, and display.

Program:

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

// Define the structure for a node in the linked list
  struct Node {
      int data;
    Struct Node * next;
    };

// Function to push an Element onto the stack
  void push (Struct Node ** top, int value) {

    // Create a new node.
Struct Node * new node
      Struct Nod == NULL) {
    if (newNode == null)
```

```
Printf ("stack overflow \n");
    return;
}

// Assign the value to the new node and
   update the top pointer.
    newNode -> data = value;
    newNode -> next = *top;

    *top = newNode;
    Printf ("%d pushed onto the Stack\n", value);
    }

// function to pop an Element from the Stack
    int pop (struct Node **top) {
        if (*top == NULL) {

    Printf ("stack underflow \n");
    return -1; //return -1 if the Stack is
                                Empty.
    }

//Get the top node, update the top pointer, and
  free the node.
  Struct Node * temp = * top;
```

Page No. _____
Date

```
int poppedvalue = temp -> data ;
*top = (*top) ->next;
free (temp);
  return poppedvalue;

}
//function to display all elements in the stack.
    void display (struct Nod * top) {
    if (top == NULL) {

    Print ("Stack is empty \n");
    return ;
  }

Struct Node * temp = top;
  Printf ("Stack element :");
    while (temp != NULL) {
    printf ("%d ", temp -> data);
      temp = temp -> next ;

  }
    printf ("\n");

    }
```

```
// Main function.
int main() {
struct Node* stack = null; // Intitialize an
                           Empty Stack.

int choice value;

while (1) {
    printf("\n stack operations :\n");
    printf ("1. push \n");
     pritt ("2. pop \n");
     printf ("3. Display \n");
     Printf ("4 Exit\n");
     Printf ("Enter your choice :");
      scanf ("%d &choice);

Switch (choice) {
    Case 1;
    // Push Operation
    Printf ("Enter the value to push :");
     Scanf ("%d", & value);
     push (2 Stack, value)
      break;
```

Page No. _____
Date

Que → 5. write a C program to implement binary tree and perform in-order, pre-order, and post-order traversals, Also, understand the Efficiency of the program.

Ans A C program that implements a binary tree and perform in-order pre-order traversals the program defines the neccessary structres to create the binary tree perform the require traversals.

Program :-

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

// Difine the structure for a node in the binary
        tree.
Struct Node {
    int data;
  Struct Node * left;
  Struct Node * right;

};
```

Page No. _____
Date

```
// function to create a new node
struct Node * createNode (int value) {
~~Stru~~
if (newnode == NULL) {
  print ("Memory allocation Failed \n");
      exit (1);
}
  newNode -> data = value;
  newNode -> left = NULL;
  newNode -> right = NULL;
    return newNode;
  }

// Function for in order traversal
  void inarder Traversal (struct Node * root) {
    if (root != NULL) {

inorder Traversal (root -> left);
  printf ("%d root -> data);
  Inorder Traversal (root -> right);

  }
}
```

Page No. _____
Date

* Efficiency Analysis :-

1. Time complexity :
   I. Each Traversal function visits each node exactly once :
   II Therefore, the time complexity of each traversal (in-order, pre-order, and post order) in O(n) wh number of nodes in the binary tree.

2. Space complexity :—
   I. The space complexity primiry depends on the recusion stack for traversal.