

PES Institute of Technology and Management

NH-206, Sagar Road, Shivamogga-577204



Department of Computer Science and Engineering

Affiliated to

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belagavi, Karnataka -590018



Laboratory Manual

III Semester B.E Data Structures Laboratory (BCSL305)

Prepared By,

**Mrs. Prathibha S,
Assistant Professor,
Department of CSE**



PES Institute of Technology and Management

Department of Computer Science and Engineering

CONTENTS

DATA STRUCTURES LABORATORY SEMESTER - III			
Course Code	BCSL305	CIE Marks	50
Number of Contact Hours/Week	0:0:2	SEEMarks	50
Total Number of Lab Contact Hours	28	ExamHours	03
Credits-1			
Course Learning Objectives:			
This laboratory course enables students to get practical experience in design, develop, implement, analyze and evaluation/testing of			
<ul style="list-style-type: none"> • Dynamic memory management • Linear data structures and their applications such as stacks, queues and lists • Non-Linear data structures and their applications such as trees and graphs 			
Descriptions (if any):			
<ul style="list-style-type: none"> • Implement all the programs in "C" Programming Language and Linux OS. 			
Programs List:			
1.	Develop a Program in C for the following: <ol style="list-style-type: none"> Declare a calendar as an array of 7 elements (A dynamically Created array) to represent 7 days of a week. Each Element of the array is a structure having three fields. The first field is the name of the Day (A dynamically allocated String), The second field is the date of the Day (A integer), the third field is the description of the activity for a particular day (A dynamically allocated String). Write functions create(), read() and display(); to create the calendar, to read the data from the keyboard and to print weeks activity details report on screen. 		
2.	Develop a Program in C for the following operations on Strings. <ol style="list-style-type: none"> Read a main String (STR), a Pattern String (PAT) and a Replace String (REP) Perform Pattern Matching Operation: Find and Replace all occurrences of PAT in STR with REP if PAT exists in STR. Report suitable messages in case PAT does not exist in STR Support the program with functions for each of the above operations. Don't use Built-in functions.		
3.	Develop a menu driven Program in C for the following operations on STACK of Integers (Array Implementation of Stack with maximum size MAX) <ol style="list-style-type: none"> Push an Element onto Stack Pop an Element from Stack Demonstrate how Stack can be used to check Palindrome Demonstrate Overflow and Underflow situations on Stack Display the status of Stack Exit Support the program with appropriate functions for each of the above operations		
4.	Develop a Program in C for converting an Infix Expression to Postfix Expression. Program should support for both parenthesized and free parenthesized Expressions with the operators: +, -, *, /, % (Remainder), ^ (Power) and alphanumeric operands.		
5.	Develop a Program in C for the following Stack Applications		



PES Institute of Technology and Management

Department of Computer Science and Engineering

	<ul style="list-style-type: none"> a. Evaluation of Suffix expression with single digit operands and operators: +, -, *, /, %, ^ b. Solving Tower of Hanoi problem with n disks
6.	<p>Develop a menu driven Program in C for the following operations on Circular QUEUE of Characters (Array Implementation of Queue with maximum size MAX)</p> <ul style="list-style-type: none"> a. Insert an Element onto CircularQUEUE b. Delete an Element from CircularQUEUE c. Demonstrate Overflow and Underflow situations on CircularQUEUE d. Display the status of CircularQUEUE e. Exit <p>Support the program with appropriate functions for each of the above operations</p>
7.	<p>Develop a menu driven Program in C for the following operations on Singly Linked List (SLL) of Student Data with the fields: <i>USN, Name, Programme, Sem, PhNo</i></p> <ul style="list-style-type: none"> a. Create a SLL of N Students Data by using <i>front insertion</i>. b. Display the status of SLL and count the number of nodes in it c. Perform Insertion/Deletion at End of SLL d. Perform Insertion/Deletion at Front of SLL (Demonstration of stack) e. Exit
8.	<p>Develop a menu driven Program in C for the following operations on Doubly Linked List (DLL) of Employee Data with the fields: <i>SSN, Name, Dept, Designation, Sal, PhNo</i></p> <ul style="list-style-type: none"> a. Create a DLL of N Employees Data by using <i>end insertion</i>. b. Display the status of DLL and count the number of nodes in it c. Perform Insertion and Deletion at End of DLL d. Perform Insertion and Deletion at Front of DLL e. Demonstrate how this DLL can be used as Double Ended Queue. f. Exit
9.	<p>Develop a Program in C for the following operations on Singly Circular Linked List (SCLL) with header nodes</p> <ul style="list-style-type: none"> a. Represent and Evaluate a Polynomial $P(x,y,z) = 6x^2y^2z - 4yz^5 + 3x^3yz + 2xy^5z - 2xyz^3$ b. Find the sum of two polynomials $POLY1(x,y,z)$ and $POLY2(x,y,z)$ and store the result in $POLYSUM(x,y,z)$ <p>Support the program with appropriate functions for each of the above operations</p>
10.	<p>Develop a menu driven Program in C for the following operations on Binary Search Tree (BST) of Integers.</p> <ul style="list-style-type: none"> a. Create a BST of N Integers: 6, 9, 5, 2, 8, 15, 24, 14, 7, 8, 5, 2 b. Traverse the BST in Inorder, Preorder and PostOrder c. Search the BST for a given element (KEY) and report the appropriate message d. Exit
11.	<p>Develop a Program in C for the following operations on Graph (G) of Cities</p> <ul style="list-style-type: none"> a. Create a Graph of N cities using Adjacency Matrix. b. Print all the nodes reachable from a given starting node in a digraph using DFS/BFS method



PES Institute of Technology and Management

Department of Computer Science and Engineering

12. Given a File of N employee records with a set K of Keys (4-digit) which uniquely determine the records in file F. Assume that file F is maintained in memory by a Hash Table (HT) of m memory locations with L as the set of memory addresses (2-digit) of locations in HT. Let the keys in K and addresses in L are Integers. Develop a Program in C that uses Hashfunction H: $K \rightarrow L$ as $H(K) = K \bmod m$ (remainder method), and implement hashing Technique to map a given key K to the address space L. Resolve the collision (if any) using linear probing.

Laboratory Outcomes:

The student should be able to:

- Analyze various linear and non-linear data structures
- Demonstrate the working nature of different types of data structures and their applications
- Use appropriate searching and sorting algorithms for the given scenario.
- Apply the appropriate data structure for solving real world problems

Conduct of Practical Examination:

- Experiment distribution
 - For laboratories having only one part: Students are allowed to pick one experiment from the lot with equal opportunity.
 - For laboratories having PART A and PART B: Students are allowed to pick one experiment from PART A and one experiment from PART B, with equal opportunity.
- Change of experiment is allowed only once and marks allotted for procedure to be made zero of the changed part only.
- Marks Distribution (*Need to change in accordance within university regulations*)
 - c) For laboratories having only one part–Procedure+Execution+Viva-Voice: $15+70+15=100$ Marks
 - d) For laboratories having PART A and PART B
 - i. Part A–Procedure + Execution +Viva = $6+28+6=40$ Marks
 - ii. Part B–Procedure + Execution +Viva = $9+42+9=60$ Marks



PES Institute of Technology and Management

Department of Computer Science and Engineering

Program 1. Develop a Program in C for the following:

- a) Declare a calendar as an array of 7 elements (A dynamically Created array) to represent 7 days of a week. Each Element of the array is a structure having three fields. The first field is the name of the Day (A dynamically allocated String), The second field is the date of the Day (A integer), the third field is the description of the activity for a particular day (A dynamically allocated String).
- b) Write functions create(), read() and display(); to create the calendar, to read the data from the keyboard and to print weeks activity details report on screen

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

struct Calendar
{
    char *name;
    int date;
    char *activity;
};

struct Calendar *C;
int i;

void create( )
{
    C=(struct Calendar *)malloc(7* sizeof(struct Calendar));
}

void read( )
{
    char buffer[20];
    for(i=0;i<7;i++)
    {
        printf("\n enter weekday");
        scanf("%s",buffer);
        (C+i)->name=(char*)malloc(strlen(buffer)+1);
        strcpy((C+i)->name,buffer);

        printf("\n Enter the date ");
        scanf("%d", &(C+i)->date);
    }
}
```



PES Institute of Technology and Management

Department of Computer Science and Engineering

```
        printf("\n Enter the activity");
        scanf("%s",buffer);
        (C+i)->activity=(char*)malloc(strlen(buffer)+1);
        strcpy((C+i)->activity,buffer);
    }

}

void display( )
{
    printf("\n Weekly Calendar of Activities\n");
    printf("-----");
    printf("\n Date\t\t\t Weekday\t\t\t Activity\n");
    printf("-----\n");
    for(i=0;i<7;i++)
    {
        printf("\n %d\t\t\t %s\t\t\t %s\n", (C+i)->date,(C+i)->name,(C+i)->activity);
    }
}

void main()
{
    create();
    read();
    display();
    free(C);

}
```

Output:

```
enter weekdayMonday
Enter the date 11
Enter the activityYoga
enter weekdayTuesday
Enter the date 12
Enter the activityYoga
enter weekdayWednesday
Enter the date 13
```



PES Institute of Technology and Management

Department of Computer Science and Engineering

Enter the activityHIT
enter weekdayThursday
Enter the date 14
Enter the activityStrengthTraining
enter weekdayFriday
Enter the date 15
Enter the activityWeightTraining
enter weekdaySaturday
Enter the date 16
Enter the activityRest
enter weekdaySunday
Enter the date 17
Enter the activityMarathonRun
Weekly Calendar of Activities

Date	Weekday	Activity
11	Monday	Yoga
12	Tuesday	Yoga
13	Wednesday	HIT
14	Thursday	StrengthTraining
15	Friday	WeightTraining
16	Saturday	Rest
17	Sunday	MarathonRun



PES Institute of Technology and Management

Department of Computer Science and Engineering

Program 2. Develop a Program in C for the following operations on Strings.

- Read a main String (STR), a Pattern String (PAT) and a Replace String (REP)
- Perform Pattern Matching Operation: Find and Replace all occurrences of PAT in STR with REP if PAT exists in STR. Report suitable messages in case PAT does not exist in STR

Support the program with functions for each of the above operations. Don't use Built-in functions.

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

char str[20], pat[20], rep[20], ans[20];
int i, j, m, k, flag=0;

void stringmatch()
{
    i = m = j = 0;
    while(str[m] != '\0')
    {
        if(str[m] == pat[i]) // ..... matching
        {
            i++; m++;
            if(pat[i] == '\0') //.....found occurrences.
            {
                flag = 1;

                for(k = 0; rep[k] != '\0'; k++, j++)
                    ans[j] = rep[k];

                i = 0;
            }
        }
        else
            ans[j++] = str[m++];
    } //while end

    ans[j] = '\0';
}
```



PES Institute of Technology and Management

Department of Computer Science and Engineering

```
void main()
{
    printf("\nEnter a main string \n");
    gets(str);
    printf("\nEnter a pattern string \n");

    gets(pat);
    printf("\nEnter a replace string \n");

    gets(rep);
    stringmatch();
    if(flag == 1)
        printf("\nThe resultant string is\n %s" , ans);
    else
        printf("\nPattern string NOT found\n");
}
```

Output:

Run1

```
Enter a main string
apple is an apple
Enter a pattern string
app
Enter a replace string
coup
The resultant string is
couple is an couple
```

Run1

```
Enter a main string
apple is an apple
Enter a pattern string
dfg
Enter a replace string
coup
Pattern string NOT found
```



PES Institute of Technology and Management

Department of Computer Science and Engineering

Program 3. Design, Develop and Implement a menu driven Program in C for the following operations on STACK of Integers (Array Implementation of Stack with maximum size MAX)

- a. Push an Element on to Stack**
- b. Pop an Element from Stack**
- c. Demonstrate how Stack can be used to check Palindrome**
- d. Demonstrate Overflow and Underflow situations on Stack**
- e. Display the status of Stack**
- f. Exit Support the program with appropriate functions for each of the above operations.**

```
#include <stdio.h>
#include<stdlib.h>
#include<math.h>
# define maxsize 5

int stack[maxsize];
int top=-1;

void push(int item)
{
    if (top==maxsize-1)
        printf("\nStack Overflow");

    else
        stack[++top]=item;
}

int pop()
{
    return stack[top--];
}

void display()
{
    int i;
    if(top== -1)
        printf("\nStack Underflow");
    else
    {
```



PES Institute of Technology and Management

Department of Computer Science and Engineering

```
        for(i=top;i>=0;i--)
            printf("\n%d",stack[i]);
    }
}

void palindrome()
{
    int num,i,rem,rev,temp;;
    printf("\n Enter a number");
    scanf("%d",&num);
    temp=num;
    while(num!=0)
    {
        rem=num%10;
        push(rem);
        num=num/10;
    }
    rev=0, i=0;
    while (top!=-1)
    {
        rev=rev+pop()*pow(10,i);
        i=i+1;
    }
    if(temp==rev)
        printf("\n No is palindrome");
    else
        printf("\n No is not a plaidrome");
}

void main()
{
    int choice, item;
    while(1)
    {
        printf("\n STACK OPERATIONS\n");
        printf("\n 1. PUSH");
        printf("\n 2.POP");
        printf("\n 3.DISPLAY");
        printf("\n 4.PALINDROME");
        printf("\n 5.EXIT");
        printf("\n \nEnter ypur choice");
        scanf("%d", &choice);
```



PES Institute of Technology and Management

Department of Computer Science and Engineering

```
switch(choice)
{
    case 1: printf("\n Enter element to pushed");
            scanf("%d",&item);
            push(item);
            break;
    case 2: if(top== -1)
            printf("\nStack Underflow");
            else
            printf("\n Element popped %d",pop());
            break;
    case 3: display();
            break;
    case 4: palindrome();
            break;
    case 5: exit(0);

    default: printf("Invalid choice");
} //switch end
} //while end
} //main end
```

Output:

STACK OPERATIONS

1. PUSH
2. POP
3. DISPLAY
4. PALINDROME
5. EXIT

Enter your choice1
Enter element to pushed11
STACK OPERATIONS

1. PUSH
2. POP
3. DISPLAY
4. PALINDROME
5. EXIT

Enter your choice1
Enter element to pushed22
STACK OPERATIONS



PES Institute of Technology and Management

Department of Computer Science and Engineering

1. PUSH
2. POP
3. DISPLAY
4. PALINDROME
5. EXIT

Enter your choice1
Enter element to pushed33
STACK OPERATIONS

1. PUSH
2. POP
3. DISPLAY
4. PALINDROME
5. EXIT

Enter your choice1
Enter element to pushed44

STACK OPERATIONS

1. PUSH
2. POP
3. DISPLAY
4. PALINDROME
5. EXIT

Enter your choice1
Enter element to pushed55
STACK OPERATIONS

1. PUSH
2. POP
3. DISPLAY
4. PALINDROME
5. EXIT

Enter your choice1
Stack Overflow
STACK OPERATIONS

1. PUSH
2. POP
3. DISPLAY
4. PALINDROME
5. EXIT

Enter your choice3
55



PES Institute of Technology and Management

Department of Computer Science and Engineering

44

33

22

11

STACK OPERATIONS

1. PUSH

2. POP

3. DISPLAY

4. PALINDROME

5. EXIT

Enter your choice

2

Element popped 55

STACK OPERATIONS

1. PUSH

2. POP

3. DISPLAY

4. PALINDROME

5. EXIT

Enter your choice2

Element popped 44

STACK OPERATIONS

1. PUSH

2. POP

3. DISPLAY

4. PALINDROME

5. EXIT

Enter your choice2

Element popped 33

STACK OPERATIONS

1. PUSH

2. POP

3. DISPLAY

4. PALINDROME

5. EXIT

Enter your choice2

Element popped 22

STACK OPERATIONS



PES Institute of Technology and Management

Department of Computer Science and Engineering

1. PUSH
2. POP
3. DISPLAY
4. PALINDROME
5. EXIT

Enter your choice2
Element popped 11
STACK OPERATIONS

1. PUSH
2. POP
3. DISPLAY
4. PALINDROME
5. EXIT
Enter your choice2
Stack Underflow
STACK OPERATIONS

1. PUSH
2. POP
3. DISPLAY
4. PALINDROME
5. EXIT

Enter your choice4
Enter a number1221
No is palindrome
STACK OPERATIONS

1. PUSH
2. POP
3. DISPLAY
4. PALINDROME
5. EXIT

Enter your choice
4
Enter a number1234
No is not a plaidrome
STACK OPERATIONS

1. PUSH
2. POP
3. DISPLAY
4. PALINDROME
5. EXIT

Enter your choice
5



PES Institute of Technology and Management

Department of Computer Science and Engineering

Program 4. Develop a Program in C for converting an Infix Expression to Postfix Expression. Program should support for both parenthesized and free parenthesized expressions with the operators: +, -, *, /, %(Remainder), ^(Power) and alphanumeric operands.

```
#include<stdio.h>
#include<stdlib.h>
#include<ctype.h>
#include<math.h>

char stack[20];
int top=-1;

void push(int item)
{
    stack[++top]=item;
}

int pop()
{
    return stack[top--];
}

int priority(char c)
{
    switch(c)
    {
        case '^':
        case '$':return 3;

        case '*':
        case '/':
        case '%':return 2;

        case '+':
        case '-':return 1;

        case '[': return -1;
    }
}
```



PES Institute of Technology and Management

Department of Computer Science and Engineering

```
void infixtopostfix()
{
    char infix[20],postfix[20];
    int i=0,j=0;
    printf("\n Enter infix expression\n");
    gets(infix);

    for(i=0;infix[i]!='\0';i++)
    {
        if(isalpha(infix[i]))
            postfix[j++]=infix[i];

        else if(infix[i]=='(')
            push(infix[i]);

        else if(infix[i]==')')
        {
            while(stack[top]!='(')
                postfix[j++]=pop();
            pop();
        }

        else if( priority(infix[i]) > priority(stack[top]))
            push(infix[i]);

        else
        {
            postfix[j++]=pop();
            push(infix[i]);
        }
    }
    while(top!=-1)
        postfix[j++]=pop();

    postfix[j]='\0';
    printf("\n\n Postfix %s",postfix);
}

void main()
{
    infixtopostfix();
}
```



PES Institute of Technology and Management

Department of Computer Science and Engineering

Output:

Enter infix expression
((a+b)*c^d)

Postfix ab+cd^*

Program 5. Design, Develop and Implement a Program in C for the following Stack Applications:

a. Evaluation of Suffix expression with single digit operands and operators: +, -, *, /, %, ^

b. Solving Tower of Hanoi problem with n disks.

a. Evaluation of Suffix expression with single digit operands and operators: +, -, *, /, %, ^

```
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
#include <math.h>
```

```
void postfixeval();
```

```
int stack[20];
int top = -1;
char postfix[20];
```

```
void push(int item)
{
    stack[++top] = item;
}
```

```
int pop()
{
    return(stack[top--]);
}
```

```
void postfixeval()
{
    int i = 0, op1, op2, result;
    printf("\n\nEnter the postfix Expression:");
```



PES Institute of Technology and Management

Department of Computer Science and Engineering

```
scanf("%s",postfix);

for(i=0;postfix[i]!='\0';i++)
{
    if(isdigit(postfix[i]))
        push(postfix[i]-'0');
    else
    {
        op2=pop();
        op1=pop();
        switch(postfix[i])
        {
            case '+':push(op1+op2);
                    break;
            case '-':push(op1-op2);
                    break;
            case '*':push(op1*op2);
                    break;
            case '/':push(op1/op2);
                    break;
            case '%':push(op1%op2);
                    break;
            case '^':push(pow(op1,op2));
                    break;
        } //switch end
    } // else end
} //for end

printf("\n Given suffix Expression n: %s\n", postfix);
printf("\nResult after Evaluation:%d\n",stack[top]);
}

void main()
{
    postfixeval();
}
```

Output:

Enter the postfix Expression: 62+5*

62+5*

Given suffix Expression n: 62+5*

Result after Evaluation:40



PES Institute of Technology and Management

Department of Computer Science and Engineering

b.Solving Tower of Hanoi problem with n disks.

```
#include <stdio.h>

void towerOfHanoi(int, char, char, char);

void main()
{
    int n; // Number of disks
    printf("Enter number of disks\n");
    scanf("%d", &n);
    towerohanoi(n, 'S', 'D', 'T');
}

// C recursive function to solve tower of hanoi puzzle

void towerofhanoi(int n, char source, char dest, char temp)
{
    if (n == 1)
    {
        printf("\n Move disk 1 from %c to %c", source, dest);
        return;
    }
    towerofhanoi(n-1, source, temp, dest);
    printf("\n Move disk %d from %c to %c", n, source, dest);
    towerofhanoi(n-1, temp, dest, source);
}
```

Output:

```
Enter number of disks
3
Move disk 1 from S to D
Move disk 2 from S to T
Move disk 1 from D to T
Move disk 3 from S to D
Move disk 1 from T to S
Move disk 2 from T to D
Move disk 1 from S to D
```



PES Institute of Technology and Management

Department of Computer Science and Engineering

Program 6. Develop a menu driven Program in C for the following operations on Circular QUEUE of Characters (Array Implementation of Queue with maximum size MAX)

- Insert an Element onto Circular QUEUE
- Delete an Element from Circular QUEUE
- Demonstrate Overflow and Underflow situations on Circular QUEUE
- Display the status of Circular QUEUE
- Exit

Support the program with appropriate functions for each of the above operations

```
#include<stdio.h>
#include<stdlib.h>
#define maxsize 5
```

```
char cq[maxsize];
int front,rear;
```

```
void insert(char item)
{
    if (front==(rear+1)% maxsize)
        printf("\n\n CIRCULAR QUEUE FULL\n");

    else
    {
        if(front==-1) //First element insertion
            front=rear=0;
        else
            rear=(rear+1)%maxsize;

        cq[rear]=item;
        printf("\nRear=%d      Front= %d\n",rear,front);
    }
}
```

```
void delete()
{
    if((front==-1))
        printf("\n CIRCULAR QUEUE EMPTY\n");
    else
    {
        printf("\nDELETED ELEMENT FROM QUEUE IS:%c\n", cq[front]);
        cq[front]= '-';
    }
}
```



PES Institute of Technology and Management

Department of Computer Science and Engineering

```
        front=(front+1)%maxsize;
        printf("\nRear = %d   Front = %d \n",rear,front);
    }
}

void display()
{
    int i;
    if((front== -1))
        printf("QUEUE EMPTY\n");
    else
    {
        printf("The Queue Elements are\n");
        for(i=0;i<maxsize;i++)
            printf("%c \t",cq[i]);
        printf("\nRear = %d Front = %d \n",rear,front);
    }
}

void main()
{
    int choice;
    char item ;
    front=-1; rear=-1;
    while(1)
    {
        printf("\n\n-----QUEUE MENU      \n");
        printf("\n1.INSERT INTO QUEUE");
        printf("\n2.DELETEF FROM QUEUE");
        printf("\n3.DISPLAY QUEUE");
        printf("\n4.EXIT");

        printf("\n\n ENTER YOUR CHOICE:");
        scanf("%d",&choice);

        switch(choice)
        {
            case 1:printf("\nENTER THEQUEUE ELEMENT:");
                    scanf(" %c",&item);
                    insert(item);
                    break;

            case 2: delete();
```



PES Institute of Technology and Management

Department of Computer Science and Engineering

```
        break;

    case 3:display();
        break;

    case 4:exit(0);

    default:printf("\nInvalidChoice.\n");
}
}
```

Output:

-----QUEUE MENU

1.INSERT INTO QUEUE
2.DELETE FROM QUEUE
3.DISPLAY QUEUE
4. EXIT

ENTER YOUR CHOICE:1
ENTER THE QUEUE ELEMENT:A
Rear=0 Front= 0

-----QUEUE MENU

1.INSERT INTO QUEUE
2.DELETE FROM QUEUE
3.DISPLAY QUEUE
4. EXIT

ENTER YOUR CHOICE:1
ENTER THE QUEUE ELEMENT:B
Rear=1 Front= 0

-----QUEUE MENU

1.INSERT INTO QUEUE
2.DELETE FROM QUEUE
3.DISPLAY QUEUE
4. EXIT



PES Institute of Technology and Management

Department of Computer Science and Engineering

ENTER YOUR CHOICE:1
ENTER THE QUEUE ELEMENT:C
Rear=2 Front= 0

-----QUEUE MENU

- 1.INSERT INTO QUEUE
- 2.DELETE FROM QUEUE
- 3.DISPLAY QUEUE
4. EXIT

ENTER YOUR CHOICE:1
ENTER THE QUEUE ELEMENT:D
Rear=3 Front= 0

-----QUEUE MENU

- 1.INSERT INTO QUEUE
- 2.DELETE FROM QUEUE
- 3.DISPLAY QUEUE
4. EXIT

ENTER YOUR CHOICE:1
ENTER THE QUEUE ELEMENT:E
Rear=4 Front= 0

-----QUEUE MENU

- 1.INSERT INTO QUEUE
- 2.DELETE FROM QUEUE
- 3.DISPLAY QUEUE
4. EXIT

ENTER YOUR CHOICE:
1
ENTER THE QUEUE ELEMENT:F
CIRCULAR QUEUE FULL

-----QUEUE MENU

- 1.INSERT INTO QUEUE



PES Institute of Technology and Management

Department of Computer Science and Engineering

2.DELETE FROM QUEUE
3.DISPLAY QUEUE
4. EXIT

ENTER YOUR CHOICE:
3

The Queue Elements are
A B C D E
Rear = 4 Front = 0

-----QUEUE MENU

1.INSERT INTO QUEUE
2.DELETE FROM QUEUE
3.DISPLAY QUEUE
4. EXIT

ENTER YOUR CHOICE:
2
DELETED ELEMENT FROM QUEUE IS: A

Rear = 4 Front = 1

-----QUEUE MENU

1.INSERT INTO QUEUE
2.DELETE FROM QUEUE
3.DISPLAY QUEUE
4. EXIT

ENTER YOUR CHOICE:
2
DELETED ELEMENT FROM QUEUE IS: B

Rear = 4 Front = 2

-----QUEUE MENU

1.INSERT INTO QUEUE
2.DELETE FROM QUEUE
3.DISPLAY QUEUE
4. EXIT



PES Institute of Technology and Management

Department of Computer Science and Engineering

ENTER YOUR CHOICE:

3

The Queue Elements are

- - C D E

Rear = 4 Front = 0

-----QUEUE MENU

1. INSERT INTO QUEUE
2. DELETE FROM QUEUE
3. DISPLAY QUEUE
4. EXIT

ENTER YOUR CHOICE:

1

ENTER THE QUEUE ELEMENT: F

Rear = 0 Front = 2

-----QUEUE MENU

1. INSERT INTO QUEUE
2. DELETE FROM QUEUE
3. DISPLAY QUEUE
4. EXIT

ENTER YOUR CHOICE:

3

The Queue Elements are

F - C D E

Rear = 0 Front = 2

-----QUEUE MENU

1. INSERT INTO QUEUE
2. DELETE FROM QUEUE
3. DISPLAY QUEUE
4. EXIT



PES Institute of Technology and Management

Department of Computer Science and Engineering

ENTER YOUR CHOICE:

1

ENTER THE QUEUE ELEMENT:C

Rear=1 Front= 2

-----QUEUE MENU

- 1.INSERT INTO QUEUE
- 2.DELETE FROM QUEUE
- 3.DISPLAY QUEUE
4. EXIT

ENTER YOUR CHOICE:

1

ENTER THE QUEUE ELEMENT:G

CIRCULAR QUEUE FULL

-----QUEUE MENU

- 1.INSERT INTO QUEUE
- 2.DELETE FROM QUEUE
- 3.DISPLAY QUEUE
4. EXIT

ENTER YOUR CHOICE: 1

ENTER THE QUEUE ELEMENT:H

CIRCULAR QUEUE FULL

-----QUEUE MENU

- 1.INSERT INTO QUEUE
- 2.DELETE FROM QUEUE
- 3.DISPLAY QUEUE
4. EXIT

ENTER YOUR CHOICE:

3



PES Institute of Technology and Management

Department of Computer Science and Engineering

The Queue Elements are

F G H D E

Rear = 1 Front = 2

-----QUEUE MENU

1. INSERT INTO QUEUE
2. DELETE FROM QUEUE
3. DISPLAY QUEUE
4. EXIT

ENTER YOUR CHOICE:

2

DELETED ELEMENT FROM QUEUE IS: D

Rear = 1 Front = 3

-----QUEUE MENU

1. INSERT INTO QUEUE
2. DELETE FROM QUEUE
3. DISPLAY QUEUE
4. EXIT

ENTER YOUR CHOICE:

2

DELETED ELEMENT FROM QUEUE IS: E

Rear = 1 Front = 4

-----QUEUE MENU

1. INSERT INTO QUEUE
2. DELETE FROM QUEUE
3. DISPLAY QUEUE
4. EXIT

ENTER YOUR CHOICE:

2

DELETED ELEMENT FROM QUEUE IS: F

Rear = 1 Front = 0

-----QUEUE MENU



PES Institute of Technology and Management

Department of Computer Science and Engineering

- 1.INSERT INTO QUEUE
- 2.DELETE FROM QUEUE
- 3.DISPLAY QUEUE
4. EXIT

ENTER YOUR CHOICE:
2

DELETED ELEMENT FROM QUEUEIS:G

Rear = 1 Front = 1

-----QUEUE MENU

- 1.INSERT INTO QUEUE
- 2.DELETE FROM QUEUE
- 3.DISPLAY QUEUE
4. EXIT

ENTER YOUR CHOICE:2

DELETED ELEMENT FROM QUEUEIS:H

Rear = -1 Front = -1
-----QUEUE MENU

- 1.INSERT INTO QUEUE
- 2.DELETE FROM QUEUE
- 3.DISPLAY QUEUE
4. EXIT

ENTER YOUR CHOICE:2

CIRCULAR QUEUE EMPTY

-----QUEUE MENU

- 1.INSERT INTO QUEUE
- 2.DELETE FROM QUEUE
- 3.DISPLAY QUEUE
4. EXIT

-----QUEUE MENU



PES Institute of Technology and Management

Department of Computer Science and Engineering

- 1.INSERT INTO QUEUE
- 2.DELETE FROM QUEUE
- 3.DISPLAY QUEUE
4. EXIT

ENTER YOUR CHOICE:

4

Program 7: Develop a menu driven Program in C for the following operations on SinglyLinkedList (SLL) of Student Data with the fields: *USN, Name, Programme, Sem, PhNo*

- a. Create a SLL of N Students Data by using *front insertion*.
- b. Display the status of SLL and count the number of nodes in it
- c. Perform Insertion/Deletion at End of SLL
- d. Perform Insertion/Deletion at Front of SLL(Demonstration of stack)
- e. Exit

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

```
struct node
{
    char usn[10];
    char name[10];
    char branch[10];
    int sem;
    char pno[10];
    struct node*next;
};
```

```
struct node * start=NULL;
struct node *temp, *p,*q;
```

```
void nodecreate()
{
    temp=(struct node*)malloc(sizeof(struct node));
```



PES Institute of Technology and Management

Department of Computer Science and Engineering

```
printf("\n EnterUSN:");
scanf("%s", temp->usn);
printf("\nEnterName:");
scanf("%s",temp->name);
printf("\nEnterbranch:");
scanf("%s",temp->branch);
printf("\nEnterSemester:");
scanf("%d",&temp->sem);
printf("\nEnterphonenumber:");
scanf("%s",temp->pno);
temp->next=NULL;
}
```

```
void insertfront()
{
    nodecreate();
    if(start==NULL)
        start=temp;
    else
    {
        temp->next=start;
        start=temp;
    }
}
```

```
void insertend()
{
    nodecreate();
    if(start==NULL)
        start=temp;
    else
    {
        q=start;
        while(q->next!=NULL)
            q=q->next;
        q->next=temp;
    }
}
```

```
void deletefront()
```



PES Institute of Technology and Management

Department of Computer Science and Engineering

```
{
    if(start==NULL)
        printf("EMPTY LIST \n");
    else
    {
        q=start;
        start=start->next;
        printf("\n Student deleted is %s ",q->usn);
        free(q);
    }
}
```

```
void deleteend()
{
    if(start==NULL)
        printf("Thelistisempty. \n");
    else
    {
        q=start;
        while(q->next!=NULL)
        {
            p=q;
            q=q->next;
        }

        p->next=NULL;
        printf("\n Student deleted is %s",q->usn);
        free(q);
    }
}
```

```
void display()
{
    if(start==NULL)
        printf("\n Listisempty.\n");
    else
    {
        printf("\n -----STUDENT DETAILS-----\n");
        for(q=start; q!=NULL;q= q->next)
            printf("%s\t%s\t%s\t%d\t%s----->",q->usn,q->name,q->branch,q->sem,q->pno);
    }
}
```



PES Institute of Technology and Management

Department of Computer Science and Engineering

```
    }  
}  
  
int main()  
{  
    int choice,n,i;  
    while(1)  
    {  
        printf("\n\n----SINGLYLINKEDLIST MENU      \n");  
        printf("1:CREATE\n");  
        printf("2:DISPLAY\n");  
        printf("3:INSERT AT END\n");  
        printf("4:INSERT AT FRONT\n");  
        printf("5:DELETE AT END\n");  
        printf("6:DELETE FROM FRONT\n");  
        printf("7:EXIT\n");  
  
        printf("Enteryourchoice\n");  
        scanf("%d",&choice);  
        switch(choice)  
        {  
            case 1:printf("Enter numberofstudents:\n");  
                    scanf("%d",&n);  
                    for(i=1;i<=n;i++)  
                        insertfront();  
                    printf("\n LIST CREATED\n");  
                    break;  
  
            case 2: display();  
                    break;  
  
            case 3:insertend();  
                    break;  
  
            case 4: insertfront();  
                    break;  
  
            case 5:deleteend();  
                    break;  
  
            case 6: deletefront();  
                    break;
```



PES Institute of Technology and Management

Department of Computer Science and Engineering

```
        case 7: exit(0);  
        default: printf("InvalidChoice\n");  
    }  
}  
}
```

Output:

```
-----SINGLYLINKEDLIST MENU  
1:CREATE  
2:DISPLAY  
3:INSERT AT END  
4:INSERT AT FRONT  
5:DELETE AT END  
6:DELETE FROM FRONT  
7:EXIT
```

```
Enter your choice  
1  
Enter number of students:  
2  
EnterUSN:100  
EnterName:AMITH  
Enterbranch:CSE  
EnterSemester:3  
Enterphonenumber:6778787  
EnterUSN:101  
EnterName:ANUSHA  
Enterbranch:ISE  
EnterSemester:4  
Enterphonenumber:65467648  
LIST CREATED
```

```
-----SINGLYLINKEDLIST MENU  
1:CREATE
```



PES Institute of Technology and Management

Department of Computer Science and Engineering

2:DISPLAY
3:INSERT AT END
4:INSERT AT FRONT
5:DELETE AT END
6:DELETE FROM FRONT
7:EXIT

Enter your choice

2

-----STUDENT DETAILS-----

101 ANUSHA ISE 4 65467648----->100AMITH CSE 3
6778787----->

-----SINGLYLINKEDLIST MENU

1:CREATE
2:DISPLAY
3:INSERT AT END
4:INSERT AT FRONT
5:DELETE AT END
6:DELETE FROM FRONT
7:EXIT

Enter your choice

3

EnterUSN:102

EnterName:ANKITHA

Enterbranch:CIVIL

EnterSemester:4

Enterphonenum:5635633

-----SINGLYLINKEDLIST MENU

1:CREATE
2:DISPLAY
3:INSERT AT END
4:INSERT AT FRONT
5:DELETE AT END
6:DELETE FROM FRONT
7:EXIT

Enter your choice

2

-----STUDENT DETAILS-----

101 ANUSHA ISE 4 65467648----->100AMITH CSE 3
6778787----->102 ANKITHA CIVIL 4 5635633----->

-----SINGLYLINKEDLIST MENU



PES Institute of Technology and Management

Department of Computer Science and Engineering

1:CREATE
2:DISPLAY
3:INSERT AT END
4:INSERT AT FRONT
5:DELETE AT END
6:DELETE FROM FRONT
7:EXIT

Enter your choice

4
EnterUSN:104
EnterName:JILL
Enterbranch:MECH
EnterSemester:4
Enterphonenumber:355667
-----SINGLYLINKEDLIST MENU

1:CREATE
2:DISPLAY
3:INSERT AT END
4:INSERT AT FRONT
5:DELETE AT END
6:DELETE FROM FRONT
7:EXIT

Enter your choice

2

-----STUDENT DETAILS-----

104	JILL	MECH	4	355667	----->	101	ANUSHA	ISE	4	65467648	---
----	>	100	AMITH	CSE	3	6778787	----->	102	ANKITHA	CIVIL	4
5635633----->											

-----SINGLYLINKEDLIST MENU

1:CREATE
2:DISPLAY
3:INSERT AT END
4:INSERT AT FRONT
5:DELETE AT END
6:DELETE FROM FRONT
7:EXIT

Enter your choice

5

Student deleted is 102

-----SINGLYLINKEDLIST MENU

1:CREATE



PES Institute of Technology and Management

Department of Computer Science and Engineering

2:DISPLAY
3:INSERT AT END
4:INSERT AT FRONT
5:DELETE AT END
6:DELETE FROM FRONT
7:EXIT
Enter your choice

6
Student deleted is 104

-----SINGLYLINKEDLIST MENU
1:CREATE
2:DISPLAY
3:INSERT AT END
4:INSERT AT FRONT
5:DELETE AT END
6:DELETE FROM FRONT
7:EXIT
Enter your choice

2
-----STUDENT DETAILS-----
101 ANUSHA ISE 4 65467648----->100AMITH CSE 3
6778787----->

-----SINGLYLINKEDLIST MENU
1:CREATE
2:DISPLAY
3:INSERT AT END
4:INSERT AT FRONT
5:DELETE AT END
6:DELETE FROM FRONT
7:EXIT
Enter your choice
7

Program 8:Develop a menu driven Program in C for the following operations on DoublyLinkedList (DLL) of Employee Data with the fields: SSN, Name, Dept, Designation,Sal ,PhNo
a. Create a DLL of N Employees Data by using end insertion.
b. Display the status of DLL and count the number of nodes in it



PES Institute of Technology and Management

Department of Computer Science and Engineering

- c. Perform Insertion and Deletion at End of DLL
- d. Perform Insertion and Deletion at Front of DLL
- e. Demonstrate how this DLL can be used as Double Ended Queue.
- f. Exit

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

```
struct node
{
    char ssn[20];
    char name[20];
    char dept[20];
    char desig[20];
    int sal;
    char pno[12];
    struct node *next;
    struct node *prev;
};
```

```
struct node *start=NULL;
struct node *temp,*q,*p;
```

```
void nodecreate()
{
    temp=(struct node *)malloc(sizeof(struct node));
    printf("Enter SSN: ");
    scanf("%s",temp->:ssn);
    printf("Enter Name: ");
    scanf("%s",temp->name);
    printf("Enter Department: ");
    scanf("%s",temp->dept);
    printf("Enter Designation: ");
    scanf("%s",temp->desig);
    printf("Enter Salary: ");
    scanf("%ld",&temp->sal);
    printf("Enter phone number: ");
    scanf("%s",temp->pno);

    temp->next=NULL;
    temp->prev=NULL;
}
```



PES Institute of Technology and Management

Department of Computer Science and Engineering

```
void insertfront()
{
    nodecreate();
    if(start==NULL)
        start=temp;
    else
    {
        temp->next=start;
        start->prev=temp;
        start=temp;
    }
}
```

```
void insertend()
{
    nodecreate();
    if(start==NULL)
        start=temp;
    else
    {
        q=start;
        while(q->next!=NULL)
            q=q->next;
        temp->prev=q;
        q->next=temp;
    }
}
```

```
void deletefront()
{
    if(start==NULL)
        printf("The DLL list is empty. \n");
    else
    {
        q=start;
        printf("Deleted Employee SSN is %s \n\n",q->ssn);
        start=start->next;
        start->prev=NULL;
        free(q);
    }
}
```



PES Institute of Technology and Management

Department of Computer Science and Engineering

```
}  
}  
  
void deleteend()  
{  
    if(start==NULL)  
        printf("Thelistisempty. \n");  
    else  
    {  
        q=start;  
        while(q->next!=NULL)  
        {  
            p=q;  
            q=q->next;  
        }  
        p->next=NULL;  
        printf("\n Student deleted is %s",q->ssn);  
        free(q);  
    }  
}  
  
void display()  
{  
    int count=0;  
    if(start==NULL)  
        printf(" DLL List is empty. \n");  
    else  
    {  
        printf("The Employee details are:\n");  
        for(q=start;q!=NULL;q=q->next)  
        {  
            printf("%s\t\t%s\t\t%s\t\t%s\t\t%d\t\t%s<=====>\n",q->ssn,q->name,q->  
dept,q->desig, q->sal,q->pno);  
            count++;  
        }  
    }  
    printf("\nTotal Number of Employee in the DLL List are: %d\n",count);  
}  
  
void main( )
```



PES Institute of Technology and Management

Department of Computer Science and Engineering

```
{
    int choice,n,i;
    while(1)
    {
        printf("\n----DOUBLY LINKED LIST MENU  \n");
        printf("1:CREATE DLL\n");
        printf("2:DISPLAY DLL\n");
        printf("DOUBLE ENDED QUEUE DEMONSTRATION \n");
        printf("3:INSERT AT END OF DLL\n");
        printf("4:DELETE FROM END OF DLL\n");
        printf("5:INSERT AT FRONT OF DLL\n");
        printf("6:DELETE FROM FRONT OF DLL\n");
        printf("7:EXIT\n");
        printf("Enter your choice\n");
        scanf("%d",&choice);
        switch(choice)
        {
            case 1:printf("Enter number of Employees:\n ");
                    scanf("%d",&n);
                    for(i=1;i<=n;i++)
                    {
                        printf("Employee %d details\n",i);
                        insertend();
                    }
                    break;
            case 2: display();
                    break;
            case 3: insertend();
                    break;
            case 4: deleteend();
                    break;
            case 5: insertfront();
                    break;
            case 6: deletefront();
                    break;
            case 7: exit(0);
            default:printf("Invalid Choice\n");
        }
    }
}
```

Output:



PES Institute of Technology and Management

Department of Computer Science and Engineering

/tmp/4frgSZyWra.o

-----DOUBLY LINKED LIST MENU

1:CREATE DLL

2:DISPLAY DLL

DOUBLE ENDED QUEUE DEMONSTRATION

3:INSERT AT END OF DLL

4:DELETE FROM END OF DLL

5:INSERT AT FRONT OF DLL

6:DELETE FROM FRONT OF DLL

7:EXIT

Enter your choice

1

Enter number of Employees:

2

Employee 1 details

Enter SSN: 11

Enter Name: Amith

Enter Department: CSE

Enter Designation: AP

Enter Salary: 45000

Enter phone number: 95876353

Employee 2 details

Enter SSN: 12

Enter Name: John

Enter Department: ISE

Enter Designation: P

Enter Salary: 89000

Enter phone number: 3436447

-----DOUBLY LINKED LIST MENU

1:CREATE DLL

2:DISPLAY DLL

DOUBLE ENDED QUEUE DEMONSTRATION

3:INSERT AT END OF DLL

4:DELETE FROM END OF DLL

5:INSERT AT FRONT OF DLL

6:DELETE FROM FRONT OF DLL

7:EXIT

Enter your choice

2

The Employee details are:

11	Amith	CSE	AP	45000 95876353<=====>
----	-------	-----	----	-----------------------

12	John	ISE	P	89000 3436447<=====>
----	------	-----	---	----------------------



PES Institute of Technology and Management

Department of Computer Science and Engineering

Total Number of Employee in the DLL List are: 2

-----DOUBLY LINKED LIST MENU

1:CREATE DLL

2:DISPLAY DLL

DOUBLE ENDED QUEUE DEMONSTRATION

3:INSERT AT END OF DLL

4:DELETE FROM END OF DLL

5:INSERT AT FRONT OF DLL

6:DELETE FROM FRONT OF DLL

7:EXIT

Enter your choice

3

Enter SSN: 13

Enter Name: Ankitha

Enter Department: EEE

Enter Designation: AP

Enter Salary: 56000

Enter phone number: 8765433

-----DOUBLY LINKED LIST MENU

1:CREATE DLL

2:DISPLAY DLL

DOUBLE ENDED QUEUE DEMONSTRATION

3:INSERT AT END OF DLL

4:DELETE FROM END OF DLL

5:INSERT AT FRONT OF DLL

6:DELETE FROM FRONT OF DLL

7:EXIT

Enter your choice

2

The Employee details are:

11	Amith	CSE	AP	45000 95876353<=====>
----	-------	-----	----	-----------------------

12	John	ISE	P	89000 3436447<=====>
----	------	-----	---	----------------------

13	Ankitha	EEE	AP	56000 8765433<=====>
----	---------	-----	----	----------------------

Total Number of Employee in the DLL List are: 3

-----DOUBLY LINKED LIST MENU

1:CREATE DLL

2:DISPLAY DLL

DOUBLE ENDED QUEUE DEMONSTRATION

3:INSERT AT END OF DLL

4:DELETE FROM END OF DLL

5:INSERT AT FRONT OF DLL



PES Institute of Technology and Management

Department of Computer Science and Engineering

6:DELETE FROM FRONT OF DLL

7:EXIT

Enter your choice

4

Student deleted is 13

-----DOUBLY LINKED LIST MENU

1:CREATE DLL

2:DISPLAY DLL

DOUBLE ENDED QUEUE DEMONSTRATION

3:INSERT AT END OF DLL

4:DELETE FROM END OF DLL

5:INSERT AT FRONT OF DLL

6:DELETE FROM FRONT OF DLL

7:EXIT

Enter your choice

5

Enter SSN: 14

Enter Name: Rihan

Enter Department: EEE

Enter Designation: P

Enter Salary: 88000

Enter phone number: 9273535

-----DOUBLY LINKED LIST MENU

1:CREATE DLL

2:DISPLAY DLL

DOUBLE ENDED QUEUE DEMONSTRATION

3:INSERT AT END OF DLL

4:DELETE FROM END OF DLL

5:INSERT AT FRONT OF DLL

6:DELETE FROM FRONT OF DLL

7:EXIT

Enter your choice

2

The Employee details are:

14	Rihan	EEE	P	88000 9273535<=====>
11	Amith	CSE	AP	45000 95876353<=====>
12	John	ISE	P	89000 3436447<=====>

Total Number of Employee in the DLL List are: 3

-----DOUBLY LINKED LIST MENU

1:CREATE DLL

2:DISPLAY DLL

DOUBLE ENDED QUEUE DEMONSTRATION



PES Institute of Technology and Management

Department of Computer Science and Engineering

3:INSERT AT END OF DLL
4:DELETE FROM END OF DLL
5:INSERT AT FRONT OF DLL
6:DELETE FROM FRONT OF DLL
7:EXIT
Enter your choice
6
Deleted Employee SSN is 14

-----DOUBLY LINKED LIST MENU
1:CREATE DLL
2:DISPLAY DLL
DOUBLE ENDED QUEUE DEMONSTRATION
3:INSERT AT END OF DLL
4:DELETE FROM END OF DLL
5:INSERT AT FRONT OF DLL
6:DELETE FROM FRONT OF DLL
7:EXIT
Enter your choice
7

Program 9: Develop a Program in C for the following operations on Singly Circular Linked List(SCLL) with header nodes

- Represent and Evaluate a Polynomial $P(x,y,z)=6x^2y^2z-4yz^5+3x^3yz+2xy^5z-2xyz^3$**
- Find the sum of two polynomials POLY1(x,y,z) and POLY2(x,y,z) and store the result in POLYSUM(x,y,z)**

Support the program with appropriate functions for each of the above operations.

Polynomial $(x,y,z)=6x^2y^2z-4yz^5+3x^3yz+2xy^5z-2xyz^3$
If $x=y=z=1$,
Result= 5

Polynomial 1= $5x^3y^3z^3+4x^2yz^5+7xy^4z^6$

Polynomial 2= $4x^3y^3z^3+8xyz$

 $9x^3y^3z^3+4x^2yz^5+7xy^4z^6+8xyz$



PES Institute of Technology and Management

Department of Computer Science and Engineering

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

struct node
{
    int coef;
    int x,y,z;
    struct node* next;
};

struct node *temp, *q;
struct node * poly,*poly1,*poly2,*polysum ;
int px,py,pz,cf;

struct node *insert(struct node *head ,int cf, int px,int py, int pz)
{
    temp = (struct node*) malloc(sizeof(struct node));
    temp->coef = cf;
    temp->x = px;
    temp->y=py;
    temp->z=pz;
    temp->next = NULL;
    if (head == NULL)
        head = temp;
    else
    {
        q=head;
        while (q->next != NULL)
            q = q->next;
        q->next = temp;
    }
    return head;
}

struct node* readpoly(struct node *head)
{
    int px,py,pz,cf;
    int n,i;
    printf("\nEnter the no of terms in polynomial ");
    scanf("%d",&n);
    for(i=1;i<=n;i++)
```



PES Institute of Technology and Management

Department of Computer Science and Engineering

```
{
    printf("\nEnter coeff: ");
    scanf("%d",&cf);
    printf("\nEnter x, y, z powers: ");
    scanf("%d%d%d", &px, &py, &pz);
    head=insert(head,cf,px,py,pz);
}
return head;
}

void print(struct node* head)
{
    for(q=head;q!=NULL; q=q->next)
        printf("%dx^%dy^%dz^%d+ ", q->coef, q->x,q->y,q->z);
}

void evaluate(struct node *head)
{
    int result=0;

    printf("\nEnter values of x, y, z");
    scanf("%d%d%d", &px, &py, &pz);
    for(q=head;q!=NULL;q=q->next)
        result = result + (q->coef * pow(px,q->x) * pow(py,q->y) * pow(pz,q->z));

    printf("\nPolynomial result is: %d", result);
}

struct node* add(struct node* poly1, struct node* poly2)
{
    struct node* polysum = NULL;

    while ((poly1 != NULL) && (poly2 != NULL))
    {
```



PES Institute of Technology and Management

Department of Computer Science and Engineering

```
if((poly1->x == poly2->x) && (poly1->y == poly2->y) &&(poly1->z ==poly1->z))
{
    polysum=insert(polysum, poly1->coef+poly2->coef,poly1->x,poly1->y,poly1->z);
    poly1 = poly1->next;
    poly2 = poly2->next;
}
else if (poly1->x > poly2->x)
{
    polysum=insert(polysum, poly1->coef, poly1->x,poly1->y,poly1->z);
    poly1 = poly1->next;
}
else
{
    polysum=insert(polysum, poly2->coef, poly2->x,poly2->y,poly2->z);
    poly2 = poly2->next;
}
}

while (poly1 != NULL)
{
    polysum=insert(polysum, poly1->coef, poly1->x,poly1->y,poly1->z);
    poly1 = poly1->next;
}
while (poly2 != NULL)
{
    polysum=insert(polysum, poly2->coef, poly2->x,poly2->y,poly2->z);
    poly2 = poly2->next;
}

return polysum;
}

void main()
{
    int choice;
    struct node *head=NULL;
    while (1)
    {
        printf("\n\n1.Read Polynomial ");
        printf("\n2. Print Polynomial");
        printf("\n3. Polynomial Evaluation");
        printf("\n4. Polynomial Additon ");
    }
}
```



PES Institute of Technology and Management

Department of Computer Science and Engineering

```
printf("\n5.Exit");
printf("\n Enter your choice");
scanf("%d",&choice);

switch(choice)
{
    case 1:printf("\nRead a Polynomial");
            head=readpoly(head);
            break;

    case 2: printf("\nPolynomial is\n");
            print(head);
            break;

    case 3: evaluate(head);
            break;

    case 4: printf("\nEnter polynomial 1 ");
            poly1=readpoly(poly1);
            printf("\nEnter polynomial 2");
            poly2=readpoly(poly2);
            printf("\n Polynomial 1\n");
            print(poly1);
            polysum=add(poly1,poly2);
            printf("\n Polynomial 2\n");
            print(poly2);
            printf("\n Polynomial Addition Result\n ");
            print(polysum);
            break;

    case 5: exit(0);

    default:printf("Invalid choice");

}
}
```

Output:

- 1.Read Polynomial
2. Print Polynomial



PES Institute of Technology and Management

Department of Computer Science and Engineering

3. Polynomial Evaluation

4. Polynomial Addition

5.Exit

Enter your choice1

Read a Polynomial

Enter the no of terms in polynomial 5

Enter coeff: 6

Enter x, y, z powers: 2

2

2

Enter coeff: -4

Enter x, y, z powers: 0

1

5

Enter coeff: 3

Enter x, y, z powers: 3

1

1

Enter coeff: 2

Enter x, y, z powers: 1

5

1

Enter coeff: -2

Enter x, y, z powers: 1

1

3

1.Read Polynomial



PES Institute of Technology and Management

Department of Computer Science and Engineering

2. Print Polynomial

3. Polynomial Evaluation

4. Polynomial Additon

5.Exit

Enter your choice2

Polynomial is

$6x^2y^2z^2 + -4x^0y^1z^5 + 3x^3y^1z^1 + 2x^1y^5z^1 + -2x^1y^1z^3 +$

1.Read Polynomial

2. Print Polynomial

3. Polynomial Evaluation

4. Polynomial Additon

5.Exit

Enter your choice3

Enter values of x, y, z1

1

1

Polynomial result is: 5

1.Read Polynomial

2. Print Polynomial

3. Polynomial Evaluation

4. Polynomial Additon

5.EXit

Enter your choice4

Enter polynomial 1

Enter the no of terms in polynomial 3

Enter coeff: 5

Enter x, y, z powers: 3

3



PES Institute of Technology and Management

Department of Computer Science and Engineering

3

Enter coeff: 4

Enter x, y, z powers: 2

1

5

Enter coeff: 7

Enter x, y, z powers: 1

4

6

Enter polynomial 2

Enter the no of terms in polynomial 2

Enter coeff: 4

Enter x, y, z powers: 3

3

3

Enter coeff: 8

Enter x, y, z powers: 1

1

1

Polynomial 1

$5x^3y^3z^3 + 4x^2y^1z^5 + 7x^1y^4z^6 +$

Polynomial 2

$4x^3y^3z^3 + 8x^1y^1z^1 +$

Polynomial Addition Result

$9x^3y^3z^3 + 4x^2y^1z^5 + 8x^1y^1z^1 + 7x^1y^4z^6 +$

Program 10: Develop a menu driven Program in C for the following operations on Binary Search Tree (BST) of Integers .



PES Institute of Technology and Management

Department of Computer Science and Engineering

- Create a BST of NIntegers:6,9,5,2,8,15,24,14,7,8,5,2
- Traverse the BST in Inorder, Preorder and PostOrder
- Search the BST for a given element(KEY)and report the appropriate message
- Exit

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

```
struct node
{
    int data;
    struct node *right,
    struct node*left;
};
```

```
struct node *root=NULL;
struct node *found;
```

```
struct node * create(struct node *root, int item)
{
    if (root == NULL)
    {
        root=(struct node *)malloc(sizeof(struct node));
        root->left = NULL;
        root->right = NULL;
        root->data = item;
    }
    else if (item < root->data)
        root->left = create(root->left, item);
    else if (item > root->data)
        root->right = create(root->right, item);
    else
        printf(" Duplicate Element Not Allowed !!!");

    return root;
}
```



PES Institute of Technology and Management

Department of Computer Science and Engineering

```
void inorder(struct node * root)
{
    if (root != NULL)
    {
        inorder(root->left);
        printf(" %d ", root->data);
        inorder(root->right);
    }
}
```

```
void preorder(struct node * root)
{
    if (root != NULL)
    {
        printf(" %d ", root->data);
        preorder(root->left);
        preorder(root->right);
    }
}
```

```
void postorder(struct node *root)
{
    if (root != NULL)
    {
        postorder(root->left);
        postorder(root->right);
        printf(" %d ", root->data);
    }
}
```

```
struct node *search(struct node* root, int key)
{
    if(root==NULL)                /* Element is not found */
        return NULL;
    if(key > root->data)            /* Search in the right sub tree. */
        return search(root->right,key);
    else if(key < root->data)      /* Search in the left sub tree. */
        return search(root->left,key);
}
```



PES Institute of Technology and Management

Department of Computer Science and Engineering

```
        else                                /* Element Found */
            return root;
    }

void main()
{
    int choice, item, n, i, key;
    while(1)
    {
        printf("\n---Binary Search Tree Operations---\n");
        printf("1-Create BST \n");
        printf("2-Inorder Traversal \n");
        printf("3-Preorder Traversal \n");
        printf("4-Postorder Traversal \n");
        printf("5-Search \n ");
        printf("6-Exit \n");

        printf("Enter your choice: ");
        scanf("%d",&choice);

        switch (choice)
        {
            case 1: printf("\n\nBST for How Many Nodes? : ");
                    scanf("%d", &n);
                    for (i=1;i<=n;i++)
                    {
                        printf("\nEnter the Data for Nodes");
                        scanf("%d",&item);
                        root=create(root, item);
                    }
                    printf("\nBST created!!\n");
                    break;

            case 2: printf("\n BST Traversal in INORDER \n");
                    inorder(root);
                    break;

            case 3: printf("\n BST Traversal in PREORDER \n");
                    preorder(root);
                    break;

            case 4: printf("\n BST Traversal in POSTORDER \n");
                    postorder(root);
                    break;
```



PES Institute of Technology and Management

Department of Computer Science and Engineering

```
case 5: printf("Enter node to be searched:\n");
        scanf("%d",&key);
        found=search(root,key);
        if(found==NULL)
        {
            printf("Search Unsuccessful.\n");
            break;
        }
        if(found->data==key)
            printf("Search Successful. Node found in BST\n");
            break;

case 6: exit(0);

default: printf("\n\nInvalid Option !!! Try Again !! \n\n");

    }
}
}
```

Output:

---Binary Search Tree Operations---

1-Create BST

2-Inorder Traversal

3-Preorder Traversal

4-Postorder Traversal

5-Search

6-Exit

Enter your choice: 1

BST for How Many Nodes? : 12

Enter the Data for Nodes6

Enter the Data for Nodes9

Enter the Data for Nodes5

Enter the Data for Nodes2



PES Institute of Technology and Management

Department of Computer Science and Engineering

Enter the Data for Nodes8

Enter the Data for Nodes15

Enter the Data for Nodes24

Enter the Data for Nodes14

Enter the Data for Nodes7

Enter the Data for Nodes8

Duplicate Element Not Allowed !!!

Enter the Data for Nodes5

Duplicate Element Not Allowed !!!

Enter the Data for Nodes

2

Duplicate Element Not Allowed !!!

BST created!!

---Binary Search Tree Operations---

1-Create BST

2-Inorder Traversal

3-Preorder Traversal

4-Postorder Traversal

5-Search

6-Exit

Enter your choice: 2

BST Traversal in INORDER

2 5 6 7 8 9 14 15 24

---Binary Search Tree Operations---

1-Create BST

2-Inorder Traversal

3-Preorder Traversal

4-Postorder Traversal

5-Search

6-Exit

Enter your choice: 3

BST Traversal in PREORDER

6 5 2 9 8 7 15 14 24

---Binary Search Tree Operations---

1-Create BST



PES Institute of Technology and Management

Department of Computer Science and Engineering

2-Inorder Traversal
3-Preorder Traversal
4-Postorder Traversal
5-Search
6-Exit
Enter your choice: 4

BST Traversal in POSTORDER
2 5 7 8 14 24 15 9 6
---Binary Search Tree Operations---
1-Create BST
2-Inorder Traversal
3-Preorder Traversal
4-Postorder Traversal
5-Search
6-Exit
Enter your choice: 5
Enter node to be searched:
Enter your choice: 5
Enter node to be searched:
14
Search Successful. Node found in BST

---Binary Search Tree Operations---
1-Create BST
2-Inorder Traversal
3-Preorder Traversal
4-Postorder Traversal
5-Search
6-Exit
Enter your choice: 5
Enter node to be searched:
3
Search Unsuccessful.

Program 11: Develop a Program in C for the following operations on Graph(G) of Cities
a) **Create a Graph of N cities using Adjacency Matrix.**
b) **Print all the nodes reachable from a given starting node in a digraph using DFS/BFS method**



PES Institute of Technology and Management

Department of Computer Science and Engineering

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

int a[10][10], n, u,v, i, j, source;
int visited[10];

void create()
{
    printf("\nEnter the number of vertices of the graph: ");
    scanf("%d", &n);
    printf("\nEnter the adjacency matrix of the graph:\n");
    for(i=0; i<n; i++)
        for(j=0; j<n; j++)
            scanf("%d", &a[i][j]);
}

void bfs(int source)
{
    int q[10], u, front=-1, rear=-1;
    visited[source] = 1;
    q[++rear] = source;
    printf("\nThe reachable vertices are: ");
    while(front<=rear)
    {
        u = q[++front];
        for(v=0; v<n; v++)
        {
            if((a[u][v] == 1)&& (visited[v] == 0))
            {
                q[++rear] = v;
                visited[v] = 1;
                printf("%d\t", v);
            }
        }
    }
}

void dfs(int v)
{
    printf("%d\t", v);
    visited[v]=1;
```



PES Institute of Technology and Management

Department of Computer Science and Engineering

```
for(i=0; i<n; i++)
{
    if((a[v][i] == 1) && (visited[i] == 0))

        dfs(i);

}
}
```

```
void main()
{
    int choice;
    while(1)
    {
        printf("\n1.Create Graph");
        printf("\n2.BFS Traversal");
        printf("\n3.DFS Traversal");
        printf("\n4.Exit");
        printf("\nEnter your choice: ");
        scanf("%d", &choice);

        switch(choice)
        {
            case 1: create();
                    break;

            case 2: for(i=0;i<n;i++)
                    visited[i]=0;
                    printf("\nEnter the source vertex : ");
                    scanf("%d", &source);
                    printf("\n Nodes that are reachable from source node %d\n", source);
                    bfs(source);
                    break;

            case 3: for(i=0;i<n;i++)
                    visited[i]=0;
                    printf("\nEnter the source vertex : ");
                    scanf("%d", &source);
                    printf("\n Nodes that are reachable from source node %d\n", source);
                    for(v=0;v<n;v++)
                    {
```



PES Institute of Technology and Management

Department of Computer Science and Engineering

```
        if((a[source][v]==1) &&(visited[v]==0))
            dfs(v);
    }

    break;

    case 4:exit(0);

    default: printf("Invalid choice");
    }
}
}
```

Output:

- 1.Create Graph
- 2.BFS Traversal
- 3.DFS Traversal
- 4.Exit

Enter your choice: 1

Enter the number of vertices of the graph: 5

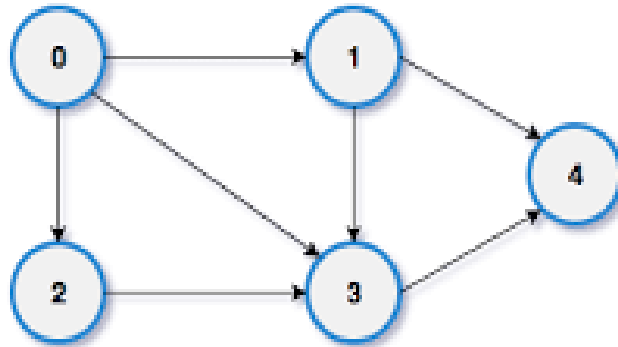
Enter the adjacency matrix of the graph:

```
0 1 1 1 0
0 0 0 1 1
0 0 0 1 0
0 0 0 0 1
0 0 0 0 0
```



PES Institute of Technology and Management

Department of Computer Science and Engineering



1.Create Graph

2.BFS Traversal

3.DFS Traversal

4.Exit

Enter your choice: 2

Enter the source vertex : 0

Nodes that are reachable from source node 0

The reachable vertices are: 1 2 3 4

1.Create Graph

2.BFS Traversal

3.DFS Traversal

4.Exit

Enter your choice: 3

Enter the source vertex : 0

Nodes that are reachable from source node 0

1 3 4 2

1.Create Graph

2.BFS Traversal

3.DFS Traversal

4.Exit

Enter your choice: 4

Program 12: Given a File of N employee records with a set K of Keys (4-digit) which



PES Institute of Technology and Management

Department of Computer Science and Engineering

uniquely determine the records in file F. Assume that file F is maintained in memory by a Hash Table (HT) of m memory locations with L as the set of memory addresses (2-digit) of locations in HT. Let the keys in K and addresses in L are Integers. Develop a Program in C that uses Hashfunction $H: K \rightarrow L$ as $H(K) = K \bmod m$ (remainder method), and implement hashing Technique to map a given key K to the address space L. Resolve the collision (if any) using linear probing.

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

#define tablesize 10

struct employee
{
    int id;
    char name[15];
    int flag;
};

struct employee emp[tablesize];
int n,i;

void create()
{
    int num, key, i;
    int ans = 1;

    for (i=0; i < tablesize; i++)
        emp[i].flag=0;

    printf("\nEnter the no of employess");
    scanf("%d",&n);
    for(i=0;i<n;i++)
    {
        printf("enter the 4 digit number id of emplyee");
        scanf("%d", &num);
        key=num % tablesize;
```



PES Institute of Technology and Management

Department of Computer Science and Engineering

```
if(emp[key].flag==0)
{
    emp[key].flag=1;
    emp[key].id=num;
    printf("enter the employee name");
    scanf("%s", &emp[key].name);
}
else
{
    while(emp[key].flag!=0)
        key=(key+1)%tablesize;

    emp[key].flag=1;
    emp[key].id=num;
    printf("enter the employee name");
    scanf("%s", &emp[key].name);
}
}
}

void display()
{
    printf("\nHashIndex\t\tEmpID\t\tEmpName");
    for(i=0; i<tablesize; i++)
        printf("\n%d\t\t\t%d\t\t%s", i, emp[i].id, emp[i].name);
}

void main()
{
    create();
    display();
}
```

Output:

```
Enter the no of employess8
enter the 4 digit number id of  employee1222
enter the employee nameAmith
enter the 4 digit number id of  employee1343
enter the employee nameAnusha
enter the 4 digit number id of  employee1333
enter the employee nameJohn
enter the 4 digit number id of  employee1505
```



PES Institute of Technology and Management

Department of Computer Science and Engineering

enter the employee nameJill
enter the 4 digit number id of employee1808
enter the employee nameKiran
enter the 4 digit number id of employee1889
enter the employee nameAmulya
enter the 4 digit number id of employee1778
enter the employee nameFiroz
enter the 4 digit number id of employee!567
enter the employee name

HashIndex	EmpID	EmpName
0	1778	Firoz
1	1778	!567
2	1222	Amith
3	1343	Anusha
4	1333	John
5	1505	Jill
6	0	
7	0	
8	1808	Kiran
9	1889	Amulya