Expt. No. 1 Date

Page No. 1

1) write a c program to find gcd numbers

```
#include<stdio.h>
#include<conio.h>
int gcd(int m, int n)
{
    if (n==0)
    return m;
    if(m<n)
        return gcd(n,m);
        return gcd(n, m%n);
}
void main()
{
    int m, n, res;
    clrscr();
    printf("enter the value of m & n");
    scanf("%d %d",&m, &n);
    res=gcd(m,n);
    printf("gcd of %d & %d is %d\n", m,
    n, res);
    getch();
}
```

Teacher's Signature

output :

Enter the value for m & n

6 8

gcd(6,8) = 2

Enter the value for m & n

6 9

gcd(6,9) = 3

Date

Expt. No. 2

Page No. 2

1) write a c programe to display pascal traingle

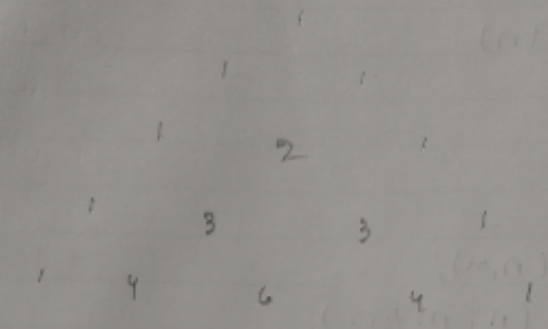
```
#include<stdio.h>
#include<conio.h>
long fact(int);
void main();
{
  int i, n, c;
  clrscr();
  printf("Enter the number of rows you wis
  h to see in pascal traingle\n");
  scanf("%d",&n);
  for(i=0; i<n; i++)
  {
    for(c=0; c<=(n-i-2); c++)
    printf(" ");
    for(c=0; c<=i; c++)
    printf("%d", fact(i)/(fact(c)*fact(i-c)));
    printf("\n");
  }
  getch();
}
long fact(int n)
{
  int c;
  long res=1;
  for(c=1; c<=n; c++)
  res= res*c;
  return res;
}
```

Teacher's Signature

output :

Enter the number of rows you wish to Pascal Triangle.

5



Date

Expt. No. 3 Page No. 3

3) write a program to generate fibonacci number

```
#include<stdio.h>
#include<conio.h>
int fib(int n)
{
  if(n==0)
   return 0;
  if(n==1)
   return 1;
  return fib(n-1)+fib(n-2);
}
void main()
{
  int i,n;
  clrscr();
  printf("Enter value of n");
  scanf("%d",&n);
  printf("%d fibbonacci number are\n",n);
  for(i=0;i<n;i++)
  {
   printf("fib(%d):%d\n",i,fib(i));
  }
  getch();
}
```

Teacher's Signature

output :-

Enter the value for n ito

10 fibonacci number are

fib(0) 0

fib(1) 1

fib(2) 1

fib(3) 2

fib(4) 3

fib(5) 5

fib(6) 8

fib(7) 13

fib(8) 21

fib(9) 34

write c program to implement tower of hanoi.

```
#include<stdio.h>
#include<conio.h>
void tower(int n, int source, int temp, int destination)
{
if(n==0)
  return;
  tower(n-1, source, destination, temp);
  printf("move disc %d FROM %c TO %c
  \n", n, source, destination);
  tower(n-1, temp, source, destination);
}
void main()
{
  int n;
  clrscr();
  printf("Enter the number of discs \n");
  scanf("%d", &n);
  tower(n, 'A', 'B', 'C');
  getch();
}
```

output:

Enter the number of discs:

3

move disc 1 from A to C

move disc 2 from A to B

move disc 1 from C to B

move disc 3 from A to C

move disc 1 from B to A

move disc 2 from B to C

move disc 1 from A to C

Write a c program to implement dynamic array find smallest & largest element of the array.

```
#include<stdio.h>
#include<conio.h>
void main()
{
int *ptr = NULL, i, n, large, small;
clrscr();
printf("Enter the size of an array:\n");
scanf("%d", &n);
ptr = (int *) malloc (n * sizeof(int));
printf("Enter the element in ");
for(i=0; i<n; i++)
{
  scanf("%d", &ptr[i]);
}
large = small = ptr[0];
for (i=1; i<n; i++)
{
  if (ptr[i] > large)
   large = ptr[i];
  if (ptr[i] < small)
   small = ptr[i];
}
printf("The smallest element is %d\n", small);
printf("The largest element is %d\n", large);
 getch();
}
```

Teacher's Signature

Output :

Enter the size of array:
5

Enter the elements
2
8
9
10
2

The smallest element is 2
The largest element is 10

Date

Expt. No. 6

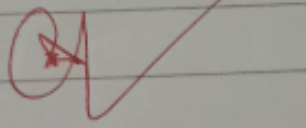
Page No. 6

6) write a c program to read the name of cities & arrange them alphabetically using bubble sort

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void sortname(char arr[10][50], int n)
{
   char temp[50];
   int i,j;
   for(i=0;i<=n-1;i++)
   {
    for(j=0;j<=n-1;j++)
    {
     if(strcmp(arr[j],arr[j+1])>0)
      {
        strcpy(temp, arr[j]);
        strcpy(arr[j], arr[j+1]);
        strcpy(arr[j+1], temp);
      }
    }
   }
}
void main()
{
   char arr[10][50];
   int n, i;
   clrscr();
   printf("Enter the number of strings\n");
   scanf("%d",&n);
```

Teacher's Signature

```
printf("Enter city name\n");
for(i=0; i<=n; i++)
fgets(arr[i], sizeof arr, stdin);
sortnames(arr, n);
printf("name in sorted order\n");
for(i=0; i<n; i++);
printf("%s", arr[i]);
getch();
}
```



Output :

Enter the number of string :

4

Enter city number :

Jade

Soraba

hasan

mysor

name in sorted order

hasan

Jade

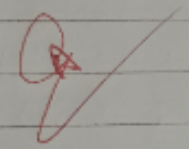
mysur

Soraba

4) write a programme to sort the given list using selection sort technique.

```
#include<stdio.h>
#include<conio.h>
void main()
{
 int n,i,j,temp,a[20],pos;
 clrscr();
 printf("Enter the number of item\n");
  scanf("%d",&n);
 printf("Enter the item to sort\n");
 for(i=0;i<n;i++)
 {
   scanf("%d",&a[i]);
 }
 for(i=0;i<n-1;i++)
 {
    pos=i;
    for(j=i+1;j<n;j++)
    {
       if(a[j]<a[pos])
       pos=j;
    }
 temp=a[pos];
 a[pos]=a[i];
 a[i]=temp;
 }
```

```
printf("The sorted item are\n");
for(i=0; i<n; i++)
{
    printf("%d\n", a[i]);
}
getch();
}
```



Output :

```
Enter the number of items
3
Enter the item to sort
4
3
2
the sorted item are
2
3
4
```

8) write a c program to sort the given list using insertion sort technique

```
#include<stdio.h>
#include<conio.h>
void insertion(int a[], int n)
{
 int i, j, item;
 for(i=1; i<n; i++)
 {
  item = a[i];
  j = i-1;
  while(item<a[j+1] && j>=0)
  {
   a[j+1] = a[j];
   j = j-1;
  }
  a[j+1] = item;
 }
}
void main()
{
 int n, j, i, item, a[20];
 clrscr();
 printf("Enter number of element\n");
 scanf("%d", &n);
 printf("Enter n element\n");
 for(i=0; i<n; i++)
 {
```

Teacher's Signature

```
    scanf("%d ",&a[i]);
}
insertion(a,n);
printf("the sorted element are \n");
for(i=0;i<n;i++)
{
  printf("%d \n",a[i]);
}
getch();
}
```

output :

Enter number of elements

4

Enter n elements

2

9

6

8

the sorted elements are

2

4

6

8

Expt. No. 1 Date

Page No. 12

PART B

1) write a c program to sort the given list using quick sort.

```
#include <stdio.h>
#include <conio.h>
int partition (int a[], int low, int high)
{
  int i, j, key, temp;
  key = a[low], i = low+1, j = high;
  while (1)
  {
    while(i<high && key >= a[i])
      i++;
    while (key<a[j])
      j--;
    if(i<j)
    temp=a[i], a[i] = a[j], a[j] = temp;
    else
    {
      temp = a[low], a[low] = a[j], a[j] = temp;
      return j;
    }
  }
}
void quicksort (int a[], int low, int high)
{
    int j;
    if (low<high)
```

Teacher's Signature

```
{
    j = partition (a, low, high);
    quicksort (a, low, j-1);
    quick sort (a, j+1, high);
  }
}
void main()
{
  int i, n, a[20];
  clrscr();
  printf("Enter the value for n : \n");
  scanf("%d", &n);
  printf("Enter the number to be sorted : \n");
  for (i=0 ; i<n ; i++);
  scanf("%d ", &a[i]);
  quicksort (a, 0, n-1);
  printf("The sorted list is : \n");
  for (i=0 ; i<n ; i++)
  printf("%d \n", a[i]);
  getch();
}
```

OUTPUT :-

Enter the number of elements to sort

5

Enter the elements to sort

90

74

76

46

35

write a c program to sort the given list using merge sort

```
#include<stdio.h>
#include<conio.h>
#define MAX 100
void merge(int a[], int low, int mid, int high)
{
 int i=low;
 int j=mid+1;
 int k=low;
 int c[MAX];
 while(i<=mid && j<=high)
 {
   if(a[i]<a[j])
  {
    c[k]=a[i];
    i=i+1;
    k=k+1;
  }
  else
  {
    c[k]=a[j];
    j=j+1;
    k=k+1;
  }
 while(i<=mid)
 {
   c[k++]=a[i++];
```

Date

Page No. 14

write a c program to sort the given list using merge sort

```
#include<stdio.h>
#include<conio.h>
#define MAX 100
void merge(int a[], int low, int mid, int high)
{
  int i=low;
  int j=mid+1;
  int k=low;
  int c[MAX];
  while(i<=mid && j<=high)
  {
    if(a[i]<a[j])
    {
      c[k]=a[i];
      i=i+1;
      k=k+1;
    }
    else
    {
      c[k]=a[j];
      j=j+1;
      k=k+1;
    }
  while(i<=mid)
  {
    c[k++]=a[i++];
```

Teacher's Signature

```
}
while (j<= high)
{
   c[k++] = a[j++];
}
for (i=low; i<= high ; i++)
{
  a[i] = c[i];
}
void merge_sort(int a[], int low, int high)
{
 int mid;
 if (low < high)
 {
  mid = (low+high)/2;
  merge_sort(a, low, high);
  merge_sort(a, mid+1, high);
  merge(a, low, mid, high);
 }
}
void main()
{
 int a[MAX], i, n;
 clrscr();
 printf("Enter the number of element to sort\n");
 scanf("%d", &n);
 printf("Enter the element to sort\n");
 for(i=0; i<n; i++)
 scanf("%d", &a[i]);
```

Teacher's Signature

```
mergesort(a, 0, n-1);
printf("The sorted list is \n");
for(i=0; i<n; i++)
printf("%d\n", a[i]);
getch();
}
```

output :

```
Enter the number of elements to sort
5
Enter the elements to sort
90
91
76
45
34

The sorted list is
34
45
76
90
91
```

Expt. No. 3 Date

Page No. 17

Write a c program to search an element using the new search & recursive binary search.

```
#include<stdio.h>
#include<conio.h>
void main()
{
int n, a[30], i, res, ch, val, item;
clrscr();
printf("\n1. Linear search \n2. Binary search \n3. Exit\n");
Printf("Enter your choice\n");
scanf("%d", &ch);
switch(ch)
{
case 1: printf("Enter the size of an array\n");
        scanf("%d", &n);
        printf("Enter the array element\n", n);
        for(i=0; i<n; i++);
        scanf("%d", &a[i]);
        Printf("Enter the key element to be
search\n");
        scanf("%d", &val);
res = linearsearch(a, n, val);
if(res == -1)
printf("element not found\n");
else
printf("element found at position %d
in ", res);
break;
```

Teacher's Signature

```
case 2: printf(" enter the the size of an array!n");
        scanf(" %d ",&n);
        printf(" Enter the element in sorted for
        made \n",n);
        for (i=0; i<n; i++)
        scanf(" %d",&a[i]);
        printf(" enter the key element to be sea
          rched \n");
        scanf(" %d",val);
        res = binarysearch (val, a, 0, n-1);
        if(res == -1)
          printf(" element not found \n");
        else
          printf(" element found at position = %d
          d \n", res);
        break;

case 3 : exit(0);
          break;
default : printf(" invalid choice\n");
}
getch();
}
int binarysearch(int a[], int n, int val)
{
  int i;
  for (i=0; i<=n; i++)
  {
    if (a[i] == val)
    return i+1;
```

```
    }
    return -1;
}
int binarysearch (int val int a[], int l,
    int h)
{
    int mid;
    if (l>h)
        return -1;
    mid = (l+h)/2;
    if (val == [mid])
    {     return mid+1;
    if (val < a[mid])
    {
        return binarysearch (val, a, l, mid-1);
    }
    else
    {
        return binarysearch (val, a, mid+1, h);
    }
}
```

output:

1. Linear search
2. Binary search
3. exit.

Enter your choice:
Enter the size of an array 3
Enter the array element
16
20
24

Enter the key element to be searched
20

Elemt found at position: 2

1. Linear search
2. Binary search
3. Exit

Enter your choice: 2
Enter the size of an array
4

Enter the element in sorted format
12
23
34
45

Enter the key element to be searched
34

element found at position. 3

4) write a c progrme to implement stacks.

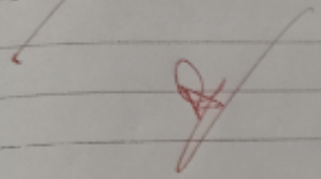
```
#include<stdio.h>
#include<conio.h>
#define STACK_SIZE 5
int top, item, item_deleted[10];
void push()
{
  if(top== STACK_SIZE-1)
  {
   printf("stack overflow");
   return ;
  }
  top= top+1;
  s[top]= item;
}
int pop()
{
  int item_deleti
  if(top==-2)
  return 0;
  item_deleted = s[top];
  top = top-1;
  return item_deleted;
}
void display()
{
  int i;
  if(top==-1)
  {
```

Teacher's Signature

```
printf("\n Stack is empty && no items to show\n");
return;
}
printf(" content of stack are \n");
for (i=0 ; i<=top ; i++)
{
  printf("%d \n", stack[i]);
}
void main()
{
  int ch;
  clrscr();
  top = -1;
  for (;;)
  {
  printf(" 1. push \n 2. pop \n 3. display \n 4. exit ");
  printf(" Enter your choice \n");
  scanf("%d", &ch);
  switch (ch)
  {
  case 1 : printf(" Enter the element to push
      ed \n");
      scanf("%d", &item);
      push();
      break;
  case 2 : item_deleted = pop();
  if (item_deleted == 0)
  printf(" Deleted element is %d \n", item
  _deleted);
  break;
```

Teacher's Signature

```
case 3 : display();
         break;
default : exit(0);
}
}
}
```



Output :-

1. push
2. pop
3. display
4. exit
enter your choice
1
Enter the elements to be pushed :
23

1. push
2. pop
3. display
4. exit
enter your choice
1
Enter the element to be pushed :
34

1. push
2. pop
3. display
4. exit
enter your choice
1
enter the element to be pushed :
45

1. push
2. pop
3. display

4. exit
enter your choice
1
enter the elements to be pushed
55

1. push
2. pop
3. display
4. exit
enter your choice
1
enter the elements to be pushed
67

1. push
2. pop
3. display
4. exit
enter your choice
1
enter the elements to be pushed
78
stack overflow

1. push
2. pop
3. display
4. exit
enter your choice
contents of stack are:
23
34
45
56
67

1. push
2. pop
3. display
4. exit
enter your choice
2
Deleted element is : 67

1. push
2. pop
3. display
4. exit
enter your choice
2
Deleted element is : 56

1. push
2. pop
3. display
4. exit
enter your choice
2
Deleted element is : 45

1. push
2. pop
3. display
4. exit

enter your choice
2

deleted element is : 23

1. push
2. pop
3. display
4. exit
enter your choice
2
deleted element is : 23

1. push
2. pop
3 display
4 . exit
enter your choice
2
stack underflow

1. push
0. pop
3. display
4. exit
enter your choice
3
stack is empty !!! no items to show

Expt. No. 5 Date

Page No. 23

5 write a c program to convert an infix expression to postfix.

```
#include<stdio.h>
#include<conio.h>
int F(char symbol)
{
  switch(symbol)
  {
    case '+':
    case '-': return 2;
    case '*':
    case '/': return 4;
    case '^':
    case '$': return 5;
    case '(': return 0;
    case '#': return -1;
    default : return 8;
  }
}
int en(char symbol)
{
  switch(ch)
  {
    case '+':
    case '-': return 1;
    case '*':
    case '/': return 3;
    case '^':
    case '$': return 6;
    case
```

Teacher's Signature

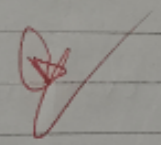
```
case '(' : return 9;
case ')' : return 0;
default : return 7;
}
}
void infix_postfix (char infix[], char postfix)
{
    int top, i, j;
    char s[30], symbol;
    top = -1;
    s[++top] = '#';
    j = 0;
for (i=0; i< strlen (infix); i++)
{
    symbol = infix [i];
    while (F(s[top]) > g(symbol))
    postfix [j++] = s[top--];
    if (F(s[top]) != g(symbol))
    s[++top] = symbol;
    else
    top --;
}
while (s[top] != '#')
postfix [j++] = s[top--];
postfix [j] = '\0';
}
void main ()
{
    char infix[20], postfix [20];
```

Teacher's Signature

Expt. No. Date

Page No. 15

```
clrscr();
printf("Enter valid infix expression\n");
scanf("%s",infix);
infix_postfix(infix,postfix);
printf("The postfix expression is\n");
printf("%s\n",postfix);
getch();
}
```



Teacher's Signature

output :

Enter valid infix expression
Cont)
the postfix expression is
abt

Expt. No. 6 Date

Page No. 26

8) Write a C program to implement simple queue.

```
#include<stdio.h>
#include<conio.h>
#define qs 4
int item, f, r, count, q[20], ch;
void insert_rear()
{
  if(r >= qs-1)
    {
    printf("queue overflow");
    return;
    }
  printf("
  r = r+1;
  q[r] = item;
}
void delete_front()
{
  if(f>r)
  {
    printf("Queue underflow");
    return;
  }
  printf("item deleted = %d\n", q[f]);
  f = f+1;
  if(f>r);
    f=0, r=-1;
}
```

Teacher's Signature

```
void display()
{
  int i;
  if(f>r)
  {
    printf("queue is empty\n");
    return;
  }
  printf("content of queue are\n");
  for(i=f; i<=r; i++)
  {
    printf("%d", q[i]);
  }
}
void main()
{
  f=0;
  r=-1;
  clrscr();
  for(;;)
  {
    printf("\n1. insert\n2. Delete\n3. Display\n4. Exit");
    printf("\nEnter your choice\n");
    scanf("%d", &ch);
    switch(ch)
    {
      case 1: printf("\nenter the item to be inser
        ted\n");
```

```
scanf("%d",&item);
insert_rear();
break;

case 2: Delete_front();
        break;

case 3: Display();
        break;

case 4: exit(0);
default: printf("wrong choice\n");
}
getch();
}
```

Output:

1. insert
2. delete
3. display
4. exit
enter your choice
1
enter the item to be inserted
11

1. insert
2. delete
3. display
4. exit
enter your choice
1
enter the item to be inserted
22

1. insert
2. delete
3. display
4. exit
enter your choice
1
enter the item to be inserted
33

1. insert
2. delete
3. display
enter your choice
1
enter the item to be inserted
44

1. insert
2. delete
3. display
4. exit
enter your choice
1
enter the item to be inserted
55
Queue Overflow

1. insert
2. delete
3. display
4. exit
enter your choice
3
contents of queue
11
22
33
44

1. insert
2. delete
3. display
4. exit
enter your choice

Item deleted = 1

1. insert
2. delete
3. display
4. exit

enter your choice
2

Item deleted = 22

1. insert
2. delete
3. display
4. exit

enter your choice
3

Item deleted = 44

1. insert
2. delete
3. display
4. exit

enter your choice
2

Queue underflow

1. insert
2. delete
3. display
4. exit

enter your choice
3

write a c program to implement linear list

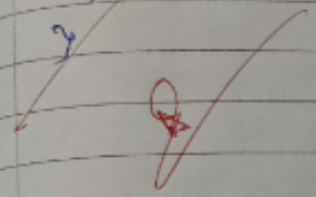
```
#include<stdio.h>
#include<conio.h>
struct node
{
    int info;
    struct node *link;
};
typedef struct node *NODE;
NODE insert_front(int, NODE);
NODE delete_front(NODE);
int display(NODE);
NODE getnode()
{
    NODE x;
    x = (NODE) malloc(sizeof(struct node));
    if(x==NULL)
    {
        printf("out of memory");
        exit(0);
    }
    return x;
}
NODE insert_front(int item, NODE first)
{
    NODE temp;
    temp = getnode();
    temp->info = item;
```

```
    temp->link = first;
    return temp;
}
NODE delete_front(NODE first)
{
    NODE temp;
    if(first == NULL)
    {
      printf("List is empty can not delete\n");
      return 0;
    }
    printf("The item deleted is %d\n", first->info);
    temp = first;
    first = temp->link;
    free(temp);
    return first;
}
int display(NODE first)
{
    NODE temp;
    if(first == NULL)
    {
      printf("List is empty\n");
      return 0;
    }
    printf("The content of the Proper linked list\n");
    temp = first;
    while(temp != NULL)
    {
```

```
        printf(" %d ", temp->info);
        temp = temp->link;
    }
    printf("\n");
    return 0;
}

void main()
{
    NODE first = NULL;
    int ch, item;
    clrscr();
    for(;;)
    {
    printf("\n1. Insert_front\n2. Delete_front\n3. Disp
          lay\n4. Exit\n");
    printf("Enter your choice\n");
    scanf("%d", &item);
    switch(ch)
    {
    case 1: printf("Enter the item to be inserted\n");
            scanf("%d", &item);
            first = insert_front(item, first);
            break;
    case 2: first = delete_front(first);
            break;
    case 3: display(first);
            break;
```

```
        case 4 : exit(0);
        default : printf("wrong choice\n");
    }
    getch();
}
```



output:

1. Insert_front
2. Delete_front
3. Display
4. Exit

Enter your choice:
1
Enter the item to be inserted
33

1. Insert_front
2. Delete_front
3. Display
4. Exit

Enter the item to be inserted
33

1. Insert_front
2. Delete_front
3 Display
4. Exit
Enter your choice: 2
Item deleted is 33

1. Insert_front
2. Delete_front
3. Display.
4. Exit
Enter your choice: 2
Item deleted is 32

1. Insert_front
2. Delete_front
3. Display
4. Exit

Enter your choice: 2

List is empty can not delete.

1. Insert_front
2. Delete_front
3 Display
4. Exit

Enter your choice:
3

List is empty

1. Insert_front
2. Delete_front
3 Display
4. Exit

Enter your choice:
3

The content of linked list

33 22

write a c program to implement traversal of a binary tree.

```
#include<stdio.h>
#include<conio.h>
struct node
{
  int info;
  struct node *lnk, *rlnk;
};
typedef struct node *NODE;
NODE getnode()
{
  NODE x;
  x= (NODE) malloc (sizeof (struct node));
  if (x== NULL)
  {
    printf("\n out of memory");
    exit(0);
  }
  return x;
}
NODE insert (int item, NODE root)
{
  NODE temp, cur, prev;
  char d[20];
  int i;
  temp = getnode();
  temp -> info= item;
```

Teacher's Signature

```
temp->llink = temp->rlink = null;
if (root == null)
return temp;
printf("\n give the direction where you want
to insert \n");
scanf("%s", d);
prev = null;
cur = root;
toupper(d);
for (i=0; i<strlen(d); i++)
{
  if (*cur = null)
  break;
  prev = cur
  if (d[i] == 'L')
  cur = cur->llink;
  else
  cur = cur->rlink;
}
if (cur_b != null || i != strlen(d))
{
  printf("\n insertion not possible");
  free(temp);
  return root;
}
if (d[i-1] == 'L')
prev->llink = temp;
else
prev->rlink = temp;
```

Teacher's Signature

```
    return root;
}
void display(NODE root, int level)
{
    int i;
    if(root==NULL)
    return;
    display(root->rlink, level+1);
    for(i=0; i<=level; i++)
    printf("  ");
    printf("%d\n", root->info);
    display(root->llink, level+1);
}
void preorder(NODE root)
{
    if(root==NULL)
    return;
    printf("%d\t", root->info);
    preorder(root->llink);
    preorder(root->rlink);
}
void postorder(NODE root)
{
    if(root==NULL)
    return;
    postorder(root->llink)
    postorder(root->rlink);
    printf("%d\t", root->info);
}
```

Teacher's Signature

```
void inorder ( NODE root)
{
  if (root == NULL)
  return;
  inorder (root->llink);
  printf("%d", root->info);
  inorder (root->rlink);
}
void main()
{
  NODE root = NULL;
  int item ch;
  clrscr();
  for (;;)
  {
  printf("\n1.insert 2. preorder\n");
  printf("3. inorder 4. postorder 5. exit\n");
  scanf("%d", &ch);
  switch (ch)
  {
  case 1: printf("\nEnter the item to be inserte
          :");
          scanf("%d", &item);
          root= insert (item, root);
          break;

case 2 : if (root == NULL)
          printf("\n tree is empty");
          else
```

```
{
printf("\n The given tree is:\n");
display (root,1);
printf("\n preorder traversal is:\n");
preorder (root);
printf("\n");
}
case 3: if (root==null)
        printf("\n tree is empty");
        else
        {
        printf("\n the given tree is:\n");
        display (root,1);
        printf("\n inorder traversal is:\n");
        inorder (root);
        printf("\n");
        }
        break;

case 4: if (root == null)
        printf("\n tree is empty");
        else
        {
        printf("\n the given tree is:\n");
        display (root,1);
        printf("\n post order traversal is:\n");
        postorder (root);
        printf("\n");
        }
```

1. insert 2. preorder 3. inorder
4. postorder 5. search 6. exit
Enter your choice : 1
Enter the item to be inserted: 1
1. insert 2. preorder 3. inorder
4. postorder 5. search 6. exit
Enter your choice : 1
Enter the item to be inserted : 5
give the direction where you want to insert : L

1. insert 2. preorder 3. inorder
4. postorder 5. search 6. exit
Enter your choice : 1
Enter the item to be inserted : 6
Give the direction where you want to insert : R

1. insert 2. preorder 3. inorder
4. postorder 5. search 6. exit
Enter your choice : 1
enter the item to be inserted : 7
give the direction where you want to inserted : LL

1. insert 2. preorder 3. inorder
4. postorder 5. search 6. exit
Enter your choice : 1
Enter the item to be inserted : 8
give the direction where you want to inserted : RR

5. inorder 6. preorder 7. inorder
4. postorder 5. search 6. exit
Enter your choice : 1
Enter the item to be inserted : 4
give the direction where you want to insert : RR

1. insert 2. preorder 3. inorder
4. postorder 5. search 6. exit
Enter your choice : 1
Enter the item to be inserted : 3
give the direction where you want to be inserted : RLL

1. insert 2. preorder 3. inorder
4. postorder 5. search 6. exit
Enter your choice : 1
Enter the item to be inserted : 2
give the direction where you want to insert-RLR

enter your choice 2

The given tree is