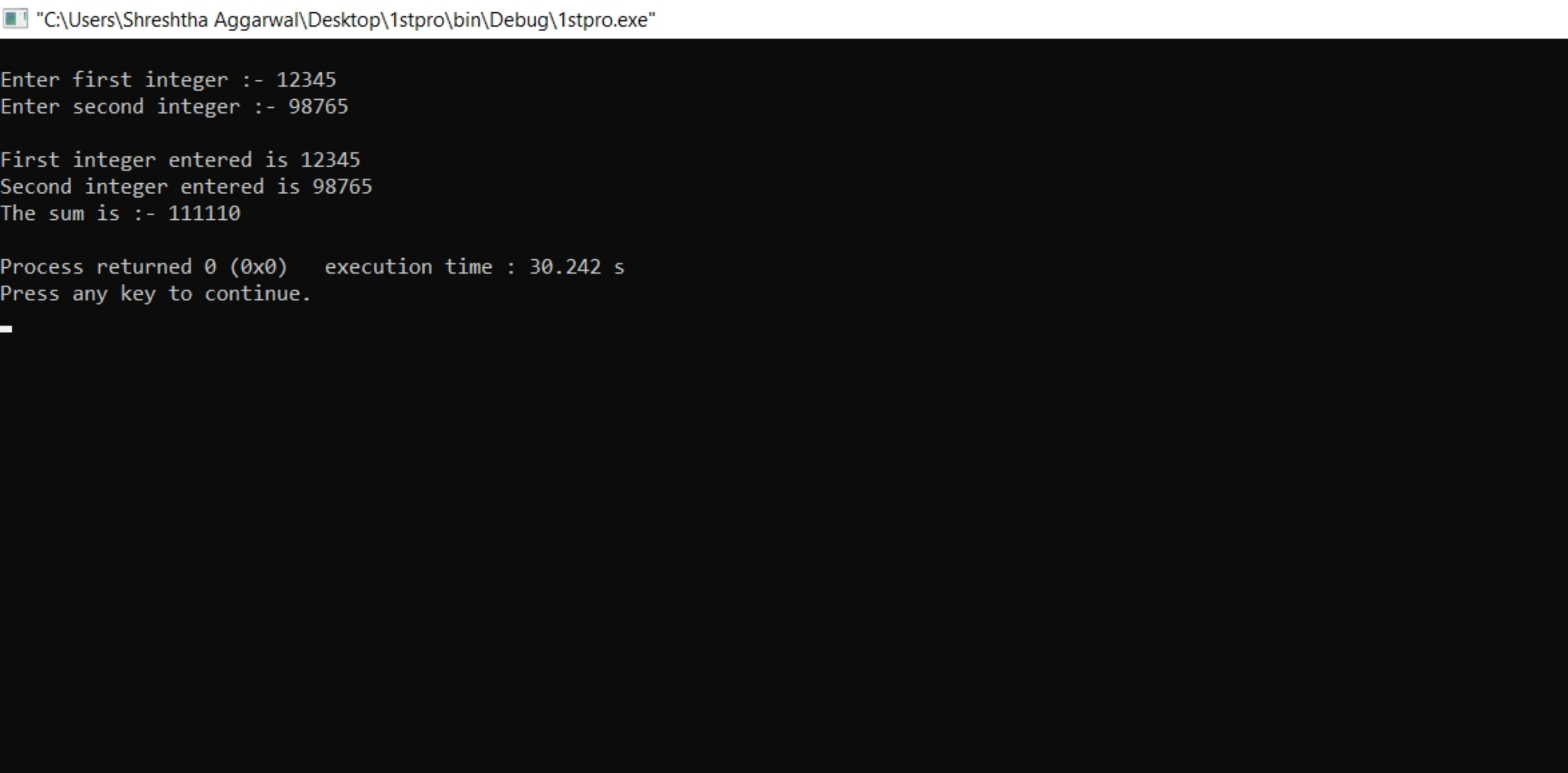
```c
//Add two large integers
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
struct NODE
{
    int info;
    struct NODE*link;
};
typedef struct NODE *node;
node getnode()
{
    node x;
    x=(node)malloc(sizeof(struct NODE));
    if(x==NULL){
        printf("Memory full..\n");
        exit(0);
    }
    return x;
}
node ins_front(node first,int item)
{
    node temp;
    temp=getnode();
    temp->info=item;
    temp->link=first;
    return temp;
}
node extract(char *s,node head)
{
    int i,n;
    for(i=0;i<strlen(s);i++){
        n=s[i]-'0';
        head=ins_front(head,n);
    }
```

```
2857        }
2858        return head;
2859    }
2860    node addlong(node head1,node head2,node head3)
2861    {
2862        int temp,sum,carry=0;
2863        node cur1,cur2;
2864        cur1=head1;
2865        cur2=head2;
2866        while(cur1!=NULL&&cur2!=NULL)
2867        {
2868            temp=cur1->info+cur2->info+carry;
2869            if(temp>9){
2870                sum=temp%10;
2871                carry=temp/10;
2872            }
2873            else{
2874                sum=temp;
2875                carry=0;
2876            }
2877            head3=ins_front(head3,sum);
2878            cur1=cur1->link;
2879            cur2=cur2->link;
2880        }
2881        while(cur1!=NULL){
2882            temp=cur1->info+carry;
2883            if(temp>9){
2884                sum=temp%10;
2885                carry=temp/10;
2886            }
2887            else{
2888                sum=temp;
2889                carry=0;
2890            }
```

```
2889                carry=0;
2890            }
2891            head3=ins_front(head3,sum);
2892            cur1=cur1->link;
2893        }
2894        while(cur2!=NULL){
2895            temp=cur2->info+carry;
2896            if(temp>9){
2897                sum=temp%10;
2898                carry=temp/10;
2899            }
2900            else{
2901                sum=temp;
2902                carry=0;
2903            }
2904            head3=ins_front(head3,sum);
2905            cur2=cur2->link;
2906        }
2907        if(cur1==NULL&&cur2==NULL){
2908            if(carry==1)
2909            head3=ins_front(head3,carry);
2910        }
2911        return head3;
2912    }
2913
2914
2915    void display(node first)
2916    {
2917        node cur;
2918        if(first==NULL){
2919            printf("Empty\n");
2920            return;
2921        }
2922        cur=first;
2923        while(cur!=NULL){
```

```
2923        while(cur!=NULL){
2924            printf("%d",cur->info);
2925            cur=cur->link;
2926        }
2927    }
2928    int main()
2929    {
2930        node head1=NULL;
2931        node head2=NULL;
2932        node head3=NULL;
2933        char s1[30],s2[30];
2934        printf("\nEnter first integer :- ");
2935        scanf("%s",s1);
2936        head1=extract(s1,head1);
2937        printf("Enter second integer :- ");
2938        scanf("%s",s2);
2939        head2=extract(s2,head2);
2940        head3=addlong(head1,head2,head3);
2941        printf("\nFirst integer entered is %s\n",s1);
2942        printf("Second integer entered is %s\n",s2);
2943        printf("The sum is :- ");
2944        display(head3);
2945        printf("\n");
2946        return 0;
2947    }
2948
```


"C:\Users\Shreshtha Aggarwal\Desktop\1stpro\bin\Debug\1stpro.exe"
Enter first integer :- 12345
Enter second integer :- 98765
First integer entered is 12345
Second integer entered is 98765
The sum is :- 111110
Process returned 0 (0x0)   execution time : 30.242 s
Press any key to continue.

```c
//Solving a Polynomial
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
#define COMPARE(x, y)((x == y)?0:(x > y)?1 : -1)
struct node
{
    float coef;
    float xexp,yexp;
    struct node *link;
};
typedef struct node *NODE;
NODE getnode()
{
    NODE t;
    t = (NODE) malloc(sizeof(struct node));
    if(t == NULL){
        printf("Memory full\n");
        return NULL;
    }
    return t;
}
NODE attach(float coef,float xexp,float yexp,NODE head)
{
    NODE temp,cur;
    temp = getnode();
    temp->coef = coef;
    temp->xexp = xexp;
    temp->yexp = yexp;
    cur = head->link;
    while(cur->link != head){
        cur = cur->link;
    }
    cur->link = temp;
```

```
2983        cur->link = temp;
2984        temp->link = head;
2985        return head;
2986    }
2987    NODE read_poly(NODE head)
2988    {
2989        int i,n;
2990        float coef,xexp,yexp;
2991        printf("\nEnter the no of terms in the polynomial :- ");
2992        scanf("%d",&n);
2993        for(i=1;i<=n;i++)
2994        {
2995            printf("\nEnter the %d term : \n",i);
2996            printf("Coefficient :- ");
2997            scanf("%f",&coef);
2998            printf("Enter power of x and y :-\n");
2999            scanf("%f",&xexp);
3000            scanf("%f",&yexp);
3001            head = attach(coef,xexp,yexp,head);
3002        }
3003          return head;
3004    }
3005    void display(NODE head)
3006    {
3007        NODE temp;
3008        if(head->link == head){
3009            printf("Polynomial does not exist.\n");
3010            return;
3011        }
3012        temp = head->link;
3013        while(temp != head){
3014            printf("%.2fx^%.2fy^%.2f",temp->coef,temp->xexp,temp->yexp);
3015            temp = temp->link;
3016            if(temp != head)
```

```
3016            if(temp != head)
3017                printf(" + ");
3018        }
3019    }
3020    float poly_evaluate(NODE head)
3021    {
3022                float x,y,sum = 0;
3023                NODE poly;
3024                printf("\nEnter the value of x and y:\n");
3025                scanf("%f%f",&x,&y);
3026                poly = head->link;
3027                while(poly != head)
3028                {
3029                            sum += poly->coef*pow(x,poly->xexp)*pow(y,poly->yexp);
3030                            poly = poly->link;
3031                }
3032                return sum;
3033    }
3034
3035    NODE poly_sum(NODE head1, NODE head2, NODE head3)
3036    {
3037        NODE a, b;
3038        float coef;
3039        a = head1->link;
3040        b = head2->link;
3041        while(a!=head1 && b!=head2){
3042            while(1){
3043                if(a->xexp == b->xexp&&a->yexp==b->yexp){
3044                    coef = a->coef + b->coef;
3045                    head3 = attach(coef,a->xexp,a->yexp,head3);
3046                    a = a->link;
3047                    b = b->link;
3048                    break;
3049                }
```

```
3049            }
3050            if(a->xexp!=0 || b->xexp!=0){
3051                switch(COMPARE(a->xexp, b->xexp)){
3052                    case -1:head3 = attach(b->coef,b->xexp,b->yexp,head3);
3053                        b = b->link;
3054                        break;
3055                    case 0:if(a->yexp > b->yexp){
3056                                head3 = attach(a->coef, a->xexp, a->yexp,head3);
3057                                a = a->link;
3058                                break;
3059                            }
3060                            else if(a->yexp < b->yexp){
3061                                head3 = attach(b->coef,b->xexp,b->yexp,head3);
3062                                b = b->link;
3063                                break;
3064                            }
3065                        case 1:head3 = attach(a->coef,a->xexp,a->yexp,head3);
3066                            a = a->link;
3067                            break;
3068                    }
3069                    break;
3070            }
3071            if(a->yexp!=0 || b->yexp!=0){
3072                switch(COMPARE(a->yexp, b->yexp)){
3073                    case -1:head3 = attach(b->coef,b->xexp,b->yexp,head3);
3074                                b = b->link;
3075                                break;
3076                    case 1:head3 = attach(a->coef,a->xexp,a->yexp,head3);
3077                            a = a->link;
3078                            break;
3079                }
3080                break;
3081            }
3082        }
```

```
3082            }
3083        }
3084        while(a!= head1){
3085            head3 = attach(a->coef,a->xexp,a->yexp,head3);
3086            a = a->link;
3087        }
3088        while(b!= head2){
3089            head3 = attach(b->coef,b->xexp,b->yexp,head3);
3090            b = b->link;
3091        }
3092        return head3;
3093    }
3094    int main()
3095    {
3096        NODE head, head1, head2, head3;
3097        int choice;
3098        float res;
3099        head =  getnode();
3100        head1 = getnode();
3101        head2 = getnode();
3102        head3 = getnode();
3103        head->link=head;
3104        head1->link=head1;
3105        head2->link=head2;
3106        head3->link= head3;
3107        for(;;){
3108            printf("\n1.Evaluate a Polynomial\n2.Evaluation of two Polynomials\n3.Exit\n");
3109            printf("Enter your choice :- ");
3110            scanf("%d",&choice);
3111            switch(choice)
3112            {
3113            case 1:printf("Enter Polynomial :- \n");
3114                head = read_poly(head);
3115                printf("Polynomial entered is :- \n");
```

```
3114            head = read_poly(head);
3115            printf("Polynomial entered is :- \n");
3116            display(head);
3117            res=poly_evaluate(head);
3118            printf("\nPolynomial value is :- %.2f\n",res);
3119            main();
3120            break;
3121        case 2:printf("Enter the first Polynomial:  \n");
3122            head1 = read_poly(head1);
3123            printf("\nPolynomial 1 is :-  \n");
3124            display(head1);
3125            printf("\nEnter the second Polynomial:  \n");
3126            head2 = read_poly(head2);
3127            printf("\nPolynomial 2 is :- \n");
3128            display(head2);
3129            printf("\nPolynomial addition result :- \n");
3130            head3 = poly_sum(head1,head2,head3);
3131            display(head3);
3132            res=poly_evaluate(head3);
3133            printf("\nPolynomial value is :- %.2f\n",res);
3134            main();
3135            break;
3136        case 3:exit(0);
3137        default:printf("Enter proper instruction!!!\n");
3138            break;
3139        }
3140    }
3141    return 0;
3142 }
```

"C:\Users\Shreshtha Aggarwal\Desktop\1stpro\bin\Debug\1stpro.exe"

```
1.Evaluate a Polynomial
2.Evaluation of two Polynomials
3.Exit
Enter your choice :- 1
Enter Polynomial :-

Enter the no of terms in the polynomial :- 2

Enter the 1 term :
Coefficient :- 1
Enter power of x and y :-
2
3

Enter the 2 term :
Coefficient :- 4
Enter power of x and y :-
5
6
Polynomial entered is :-
1.00x^2.00y^3.00 + 4.00x^5.00y^6.00
Enter the value of x and y:
1
2

Polynomial value is :- 264.00

1.Evaluate a Polynomial
2.Evaluation of two Polynomials
3.Exit
Enter your choice :- 2
Enter the first Polynomial:

Enter the no of terms in the polynomial :- 2

Enter the 1 term :
Coefficient :- 1
Enter power of x and y :-
2
3

Enter the 2 term :
Coefficient :- 2
Enter power of x and y :-
3
1

Polynomial 1 is :-
1.00x^2.00y^3.00 + 2.00x^3.00y^1.00
```

"C:\Users\Shreshtha Aggarwal\Desktop\1stpro\bin\Debug\1stpro.exe"

```
Polynomial 1 is :-
1.00x^2.00y^3.00 + 2.00x^3.00y^1.00
Enter the second Polynomial:

Enter the no of terms in the polynomial :- 2

Enter the 1 term :
Coefficient :- 4
Enter power of x and y :-
2
3

Enter the 2 term :
Coefficient :- 5
Enter power of x and y :-
1
4

Polynomial 2 is :-
4.00x^2.00y^3.00 + 5.00x^1.00y^4.00
Polynomial addition result :-
5.00x^2.00y^3.00 + 2.00x^3.00y^1.00 + 5.00x^1.00y^4.00
Enter the value of x and y:
1
2

Polynomial value is :- 124.00

1.Evaluate a Polynomial
2.Evaluation of two Polynomials
3.Exit
Enter your choice :- 3

Process returned 0 (0x0)   execution time : 58.308 s
Press any key to continue.
```

```c
//Binary Search Tree
#include <stdio.h>
#include <stdlib.h>
struct node {
  int info;
  struct node *llink,*rlink;
};typedef struct node *Node;
Node getnode(int item) {
  Node temp = (Node )malloc(sizeof(struct node));
  temp->info = item;
  temp->llink = temp->rlink = NULL;
  return temp;
}
Node insert(Node node, int info) {
  if (node == NULL)
    return getnode(info);
  if (info < node->info)
    node->llink = insert(node->llink, info);
  else
    node->rlink = insert(node->rlink, info);
  return node;
}
void preorder(Node root) {
  if(root == NULL){
    return;
  }
  printf("%d -> ",root->info);
  preorder(root->llink);
  preorder(root->rlink);
}
void inorder(Node root) {
  if(root == NULL){
    return;
  }
  inorder(root->llink);
```

```
3337    inorder(root->llink);
3338    printf("%d -> ",root->info);
3339    inorder(root->rlink);
3340  }
3341  void postorder(Node root) {
3342    if(root == NULL){
3343      return;
3344    }
3345    postorder(root->llink);
3346    postorder(root->rlink);
3347    printf("%d -> ",root->info);
3348  }
3349  Node delete(Node root,int item)
3350  {
3351    Node cur,parent,q,suc;
3352    if(root==NULL)
3353    {
3354      printf("Empty tree\n");
3355      return root;
3356    }
3357    parent=NULL;
3358    cur=root;
3359    while(cur!=NULL&&item!=cur->info)
3360    {
3361      parent=cur;
3362      cur=(item<cur->info)?cur->llink:cur->rlink;
3363    }
3364    if(cur==NULL)
3365    {
3366      printf("Element not found\n");
3367      return root;
3368    }
3369    if(cur->llink==NULL)
3370      q=cur->rlink;
```

```
3370      q=cur->rlink;
3371    else if(cur->rlink==NULL)
3372      q=cur->llink;
3373    else
3374    {
3375      suc=cur->rlink;
3376      while(suc->llink!=NULL)
3377        suc=suc->llink;
3378      suc->llink=cur->llink;
3379      q=cur->rlink;
3380    }
3381    if(parent==NULL)
3382      return q;
3383    if(cur==parent->llink)
3384      parent->llink=q;
3385    else
3386      parent->rlink=q;
3387    free(cur);
3388    return root;
3389  }
3390  void display(Node root,int i)
3391  {
3392    int j;
3393    if(root!=NULL)
3394    {
3395      display(root->rlink,i+1);
3396      for(j=0;j<i;j++)
3397        printf("   ");
3398      printf("%d\n",root->info);
3399      display(root->llink,i+1);
3400    }
3401  }
3402  int main() {
3403    Node root = NULL;
```

```
int main() {
  Node root = NULL;
  int choice,item;
  for(;;){
    printf("\n1.Insert\n2.Preorder\n3.Inorder\n4.Postorder\n5.Delete\n6.Display\n7.exit\n");
    printf("Enter choice : ");
    scanf("%d",&choice);
    switch(choice){
      case 1:printf("Enter item to be inserted : ");
             scanf("%d",&item);
             root = insert(root,item);
             break;
      case 2:printf("Preorder traversal: ");
          preorder(root);
             break;
      case 3:printf("Inorder traversal: ");
             inorder(root);
             break;
      case 4:printf("Postorder traversal: ");
             postorder(root);
             break;
      case 5:printf("Enter the item : ");
             scanf("%d",&item);
             root=delete(root,item);
             break;
      case 6:display(root,0);
             break;
      case 7:exit(0);
      default:printf("Enter proper instructions!!\n");
             break;
    }
  }
  return 0;
}
```

"C:\Users\Shreshtha Aggarwal\Desktop\1stpro\bin\Debug\1stpro.exe"

```
1.Insert
2.Preorder
3.Inorder
4.Postorder
5.Delete
6.Display
7.exit
Enter choice : 1
Enter item to be inserted : 20

1.Insert
2.Preorder
3.Inorder
4.Postorder
5.Delete
6.Display
7.exit
Enter choice : 1
Enter item to be inserted : 10

1.Insert
2.Preorder
3.Inorder
4.Postorder
5.Delete
6.Display
7.exit
Enter choice : 1
Enter item to be inserted : 5

1.Insert
2.Preorder
3.Inorder
4.Postorder
5.Delete
6.Display
7.exit
Enter choice : 1
Enter item to be inserted : 13

1.Insert
2.Preorder
3.Inorder
4.Postorder
5.Delete
6.Display
7.exit
Enter choice : 1
Enter item to be inserted : 30
```

"C:\Users\Shreshtha Aggarwal\Desktop\1stpro\bin\Debug\1stpro.exe"

```
1.Insert
2.Preorder
3.Inorder
4.Postorder
5.Delete
6.Display
7.exit
Enter choice : 1
Enter item to be inserted : 24

1.Insert
2.Preorder
3.Inorder
4.Postorder
5.Delete
6.Display
7.exit
Enter choice : 1
Enter item to be inserted : 39

1.Insert
2.Preorder
3.Inorder
4.Postorder
5.Delete
6.Display
7.exit
Enter choice : 6
    39
  30
    24
20
    13
  10
    5

1.Insert
2.Preorder
3.Inorder
4.Postorder
5.Delete
6.Display
7.exit
Enter choice : 2
Preorder traversal: 20 -> 10 -> 5 -> 13 -> 30 -> 24 -> 39 ->
1.Insert
2.Preorder
3.Inorder
4.Postorder
```

"C:\Users\Shreshtha Aggarwal\Desktop\1stpro\bin\Debug\1stpro.exe"

```
5.Delete
6.Display
7.exit
Enter choice : 3
Inorder traversal: 5 -> 10 -> 13 -> 20 -> 24 -> 30 -> 39 ->
1.Insert
2.Preorder
3.Inorder
4.Postorder
5.Delete
6.Display
7.exit
Enter choice : 4
Postorder traversal: 5 -> 13 -> 10 -> 24 -> 39 -> 30 -> 20 ->
1.Insert
2.Preorder
3.Inorder
4.Postorder
5.Delete
6.Display
7.exit
Enter choice : 5
Enter the item : 39

1.Insert
2.Preorder
3.Inorder
4.Postorder
5.Delete
6.Display
7.exit
Enter choice : 6
  30
    24
20
    13
  10
    5

1.Insert
2.Preorder
3.Inorder
4.Postorder
5.Delete
6.Display
7.exit
Enter choice : 7

Process returned 0 (0x0)   execution time : 70.263 s
Press any key to continue.
```