

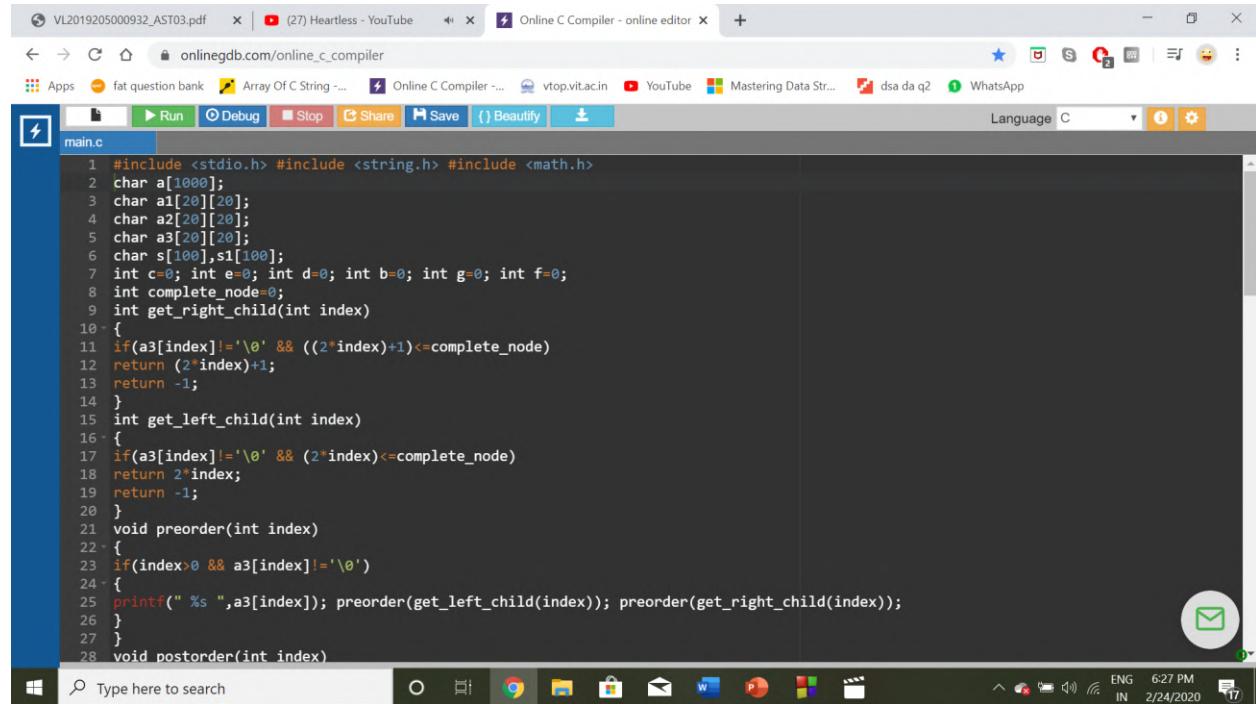
Name: Kulvir Singh

Reg No.:19BCE2074

Slot: B1

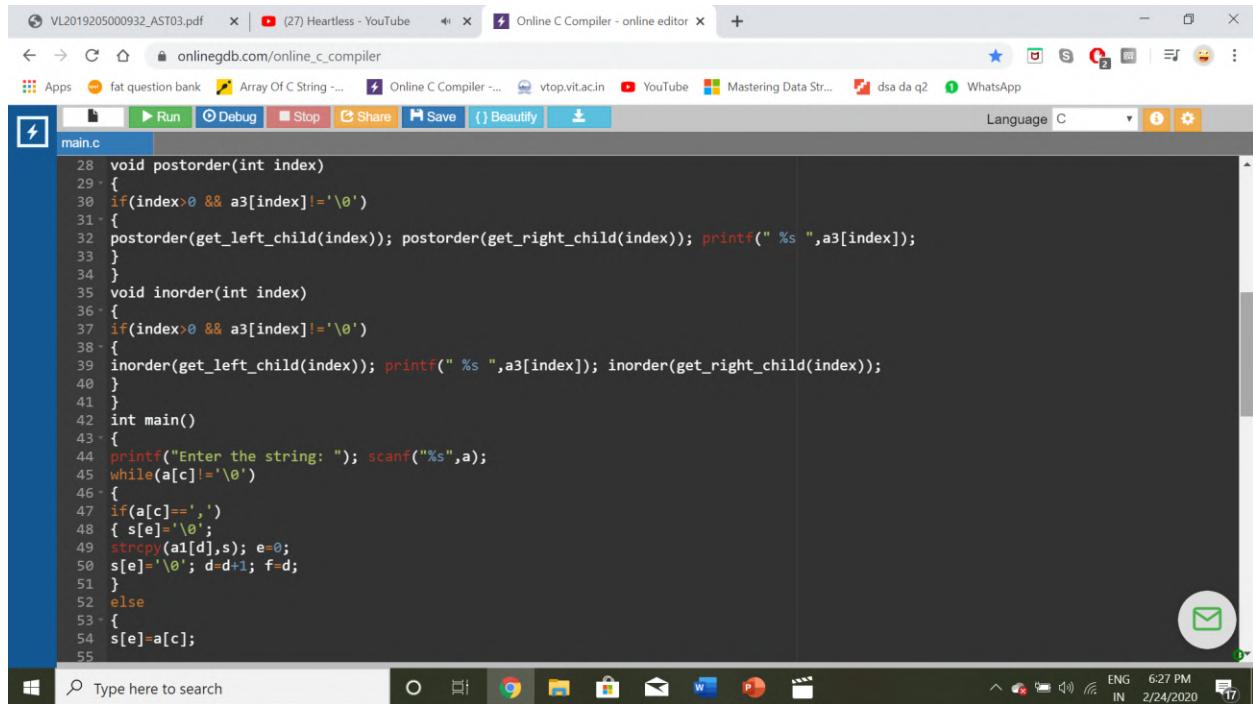
DSA DA 3

Question 1



The screenshot shows a web-based C compiler interface. The code editor contains a C program named 'main.c' which defines functions for preorder and postorder traversal of a binary tree represented by arrays a1, a2, and a3. The code includes comments explaining the logic for finding left and right children based on the value at index 3.

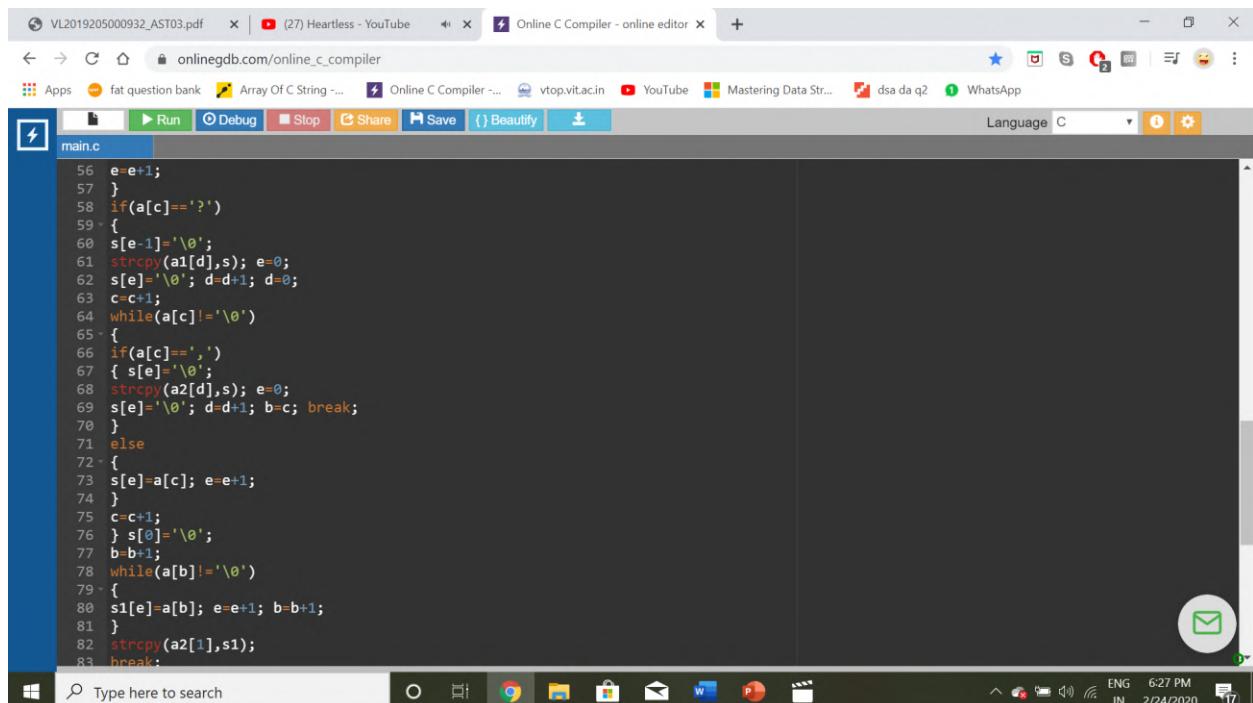
```
#include <stdio.h> #include <string.h> #include <math.h>
char a1[100];
char a1[20][20];
char a2[20][20];
char a3[20][20];
char s[100],s1[100];
int c=0; int e=0; int d=0; int b=0; int g=0; int f=0;
int complete_node=0;
int get_right_child(int index)
{
    if(a3[index]!='\0' && ((2*index)+1)<=complete_node)
        return (2*index)+1;
    return -1;
}
int get_left_child(int index)
{
    if(a3[index]!='\0' && (2*index)<=complete_node)
        return 2*index;
    return -1;
}
void preorder(int index)
{
    if(index>0 && a3[index]!='\0')
    {
        printf(" %s ",a3[index]); preorder(get_left_child(index)); preorder(get_right_child(index));
    }
}
void postorder(int index)
```



VL2019205000932_AST03.pdf | (27) Heartless - YouTube | Online C Compiler - online editor

main.c

```
28 void postorder(int index)
29 {
30     if(index>0 && a3[index]!='\0')
31     {
32         postorder(get_left_child(index)); postorder(get_right_child(index)); printf(" %s ",a3[index]);
33     }
34 }
35 void inorder(int index)
36 {
37     if(index>0 && a3[index]!='\0')
38     {
39         inorder(get_left_child(index)); printf(" %s ",a3[index]); inorder(get_right_child(index));
40     }
41 }
42 int main()
43 {
44     printf("Enter the string: "); scanf("%s",a);
45     while(a[c]!='\0')
46     {
47         if(a[c]==',')
48         { s[e]='\0';
49             strcpy(a1[d],s); e=0;
50             s[e]='\0'; d=d+1; f=d;
51         }
52         else
53         {
54             s[e]=a[c];
55             e=e+1;
56         }
57     }
58     if(a[c]=='?')
59     {
60         s[e-1]='\0';
61         strcpy(a1[d],s); e=0;
62         s[e]='\0'; d=d+1; d=0;
63         c=c+1;
64         while(a[c]!='\0')
65         {
66             if(a[c]==',')
67             { s[e]='\0';
68                 strcpy(a2[d],s); e=0;
69                 s[e]='\0'; d=d+1; b=c; break;
70             }
71             else
72             {
73                 s[e]=a[c]; e=e+1;
74             }
75             c=c+1;
76         } s[0]='\0';
77         b=b+1;
78         while(a[b]!='\0')
79         {
80             s1[e]=a[b]; e=e+1; b=b+1;
81         }
82         strcpy(a2[1],s1);
83         break;
84     }
85 }
```



VL2019205000932_AST03.pdf | (27) Heartless - YouTube | Online C Compiler - online editor

main.c

```
56 e=e+1;
57 }
58 if(a[c]=='?')
59 {
60 s[e-1]='\0';
61 strcpy(a1[d],s); e=0;
62 s[e]='\0'; d=d+1; d=0;
63 c=c+1;
64 while(a[c]!='\0')
65 {
66     if(a[c]==',')
67     { s[e]='\0';
68         strcpy(a2[d],s); e=0;
69         s[e]='\0'; d=d+1; b=c; break;
70     }
71     else
72     {
73         s[e]=a[c]; e=e+1;
74     }
75     c=c+1;
76 } s[0]='\0';
77 b=b+1;
78 while(a[b]!='\0')
79 {
80     s1[e]=a[b]; e=e+1; b=b+1;
81 }
82 strcpy(a2[1],s1);
83 break;
84 }
```

VL2019205000932_AST03.pdf | (27) Heartless - YouTube | Online C Compiler - online editor

main.c

```
79: {  
80:     s1[e]=a[b]; e=e+1; b=b+1;  
81: }  
82: strcpy(a2[1],s1);  
83: break;  
84: }  
85: c=c+1;  
86: }  
87: if(pow(2,floor(log(f+1))==f+1))  
88: {complete_node=f+1;  
89: else complete_node=pow(2,floor(log(f+1))); int i;  
90: for(i=0;i<f+1;i++)  
91: {  
92:     if(strcmp(a1[i],a2[0])==0)  
93:     {  
94:         c=i;  
95:     }  
96:     if(strcmp(a1[i],a2[1])==0)  
97:     {  
98:         g=i;  
99:     }  
100:    strcpy(a3[i+1],a1[i]);  
101: }  
102: printf("Preorder:\n"); preorder(1); printf("\nPostorder:\n"); postorder(1); printf("\nInorder:\n"); inorder(1);  
103: printf("\n");  
104: return 0;  
105: }  
106:
```

VL2019205000932_AST03.pdf | (27) Heartless - YouTube | Online C Compiler - online editor

main.c:49:1: note: include <string.h> or provide a declaration of 'strcpy'
main.c:61:1: warning: incompatible implicit declaration of built-in function 'strcpy'
main.c:61:1: note: include <string.h> or provide a declaration of 'strcpy'
main.c:87:4: warning: implicit declaration of function 'pow' [-Wimplicit-function-declaration]
main.c:87:4: warning: incompatible implicit declaration of built-in function 'pow'
main.c:87:4: note: include <math.h> or provide a declaration of 'pow'
main.c:87:10: warning: implicit declaration of function 'floor' [-Wimplicit-function-declaration]
main.c:87:10: warning: incompatible implicit declaration of built-in function 'floor'
main.c:87:10: note: include <math.h> or provide a declaration of 'floor'
main.c:87:16: warning: implicit declaration of function 'log' [-Wimplicit-function-declaration]
main.c:87:16: warning: incompatible implicit declaration of built-in function 'log'
main.c:87:16: note: include <math.h> or provide a declaration of 'log'
main.c:92:4: warning: implicit declaration of function 'strcmp' [-Wimplicit-function-declaration]
main.c:100:1: warning: incompatible implicit declaration of built-in function 'strcpy'
main.c:100:1: note: include <string.h> or provide a declaration of 'strcpy'
Enter the string: kulvir,kawal,aksh,vinay,manuj,gagan,prabh,arjun
Preorder:
kulvir kawal vinay manuj aksh gagan prabh
Postorder:
vinay manuj kawal gagan prabh aksh kulvir
Inorder:
vinay kawal manuj kulvir gagan aksh prabh

...Program finished with exit code 0
Press ENTER to exit console

Question 2

Online C Compiler - online editor | VL2019205000932_AST03.pdf | Online C Compiler - online editor

onlinegdb.com/online_c_compiler

main.cpp

```
1 #include<iostream>
2 #include<stdlib.h>
3 using namespace std;
4 struct node{
5     int data;
6     struct node *left;
7     struct node *right;
8 };
9 struct node *newNode(int data)
10 {
11     struct node *temp=(struct node*)malloc(sizeof(struct node));
12     temp->data=data;
13     temp->left=NULL;
14     temp->right=NULL;
15     return temp;
16 }
17 void insert_node(struct node*root,int n1,int n2,char lr)
18 {
19     if(root==NULL)
20         return;
21     if(root->data==n1)
22     {
23         switch(lr)
24         {
25             case 'l':root->left=newNode(n2);
26             break;
27             case 'r':root->right=newNode(n2);
28             break;
29         }
30     }
31     else
32     {
33         insert_node(root->left,n1,n2,lr);
34         insert_node(root->right,n1,n2,lr);
35     }
36 }
```

Type here to search

1:27 PM 2/24/2020

Online C Compiler - online editor | VL2019205000932_AST03.pdf | Online C Compiler - online editor

onlinegdb.com/online_c_compiler

main.cpp

```
25             case 'l':root->left=newNode(n2);
26             break;
27             case 'r':root->right=newNode(n2);
28             break;
29     }
30 }
31 else
32 {
33     insert_node(root->left,n1,n2,lr);
34     insert_node(root->right,n1,n2,lr);
35 }
36 }
37 void inorder(struct node *root)
38 {
39     if(root==NULL)
40         return;
41     inorder(root->left);
42     cout<<root->data<<" ";
43     inorder(root->right);
44 }
45 }
46 void preorder(struct node *root)
47 {
48     if(root==NULL)
49         return;
50     cout<<root->data<<" ";
51     preorder(root->left);
52 }
```

Type here to search

1:27 PM 2/24/2020

Online C Compiler - online editor | VL2019205000932_AST03.pdf | Online C Compiler - online editor

onlinegdb.com/online_c_compiler

Apps fat question bank Array Of C String ... Online C Compiler ... vtop.vit.ac.in YouTube Mastering Data Str... dsa da q2 WhatsApp

main.cpp

```
49     return;
50     cout<<root->data<<"";
51     preorder(root->left);
52     preorder(root->right);
53 }
54 void postorder(struct node *root)
55 {
56     if(root==NULL)
57         return;
58     postorder(root->left);
59     postorder(root->right);
60     cout<<root->data<<"";
61 }
62 int main()
63 {
64     struct node *root=NULL;
65     int n;
66     cout<<"\nenter the number of edges: ";
67     cin>>n;
68     cout<<"\ninput the nodes of the binary tree in order\n\nparent-child-left(or)right-\n\n";
69     while(n--)
70     {
71         char lr;
72         int n1,n2;
73         cin>>n1>>n2;
74         cin>>lr;
75         if(root==NULL)
```

Language C++ Run Debug Stop Share Save Beautify

input stderr

Type here to search

Online C Compiler - online editor | VL2019205000932_AST03.pdf | Online C Compiler - online editor

onlinegdb.com/online_c_compiler

Apps fat question bank Array Of C String ... Online C Compiler ... vtop.vit.ac.in YouTube Mastering Data Str... dsa da q2 WhatsApp

main.cpp

```
75
76     if(root==NULL)
77     {
78         root=newNode(n1);
79         switch(lr);
80         {
81             case 'l':root->left=newNode(n2);
82                 break;
83             case 'r':root->right=newNode(n2);
84                 break;
85         }
86     else
87     {
88         insert_node(root,n1,n2,lr);
89     }
90 }
91 cout<<"\ninorder traversal:";
92 inorder(root);
93 cout<<endl;
94 cout<<"\npreorder traversal:";
95 preorder(root);
96 cout<<endl;
97 cout<<"\npostorder traversal:";
98 postorder(root);
99 cout<<endl;
100 return 0;
101 }
```

Language C++ Run Debug Stop Share Save Beautify

input stderr

Type here to search

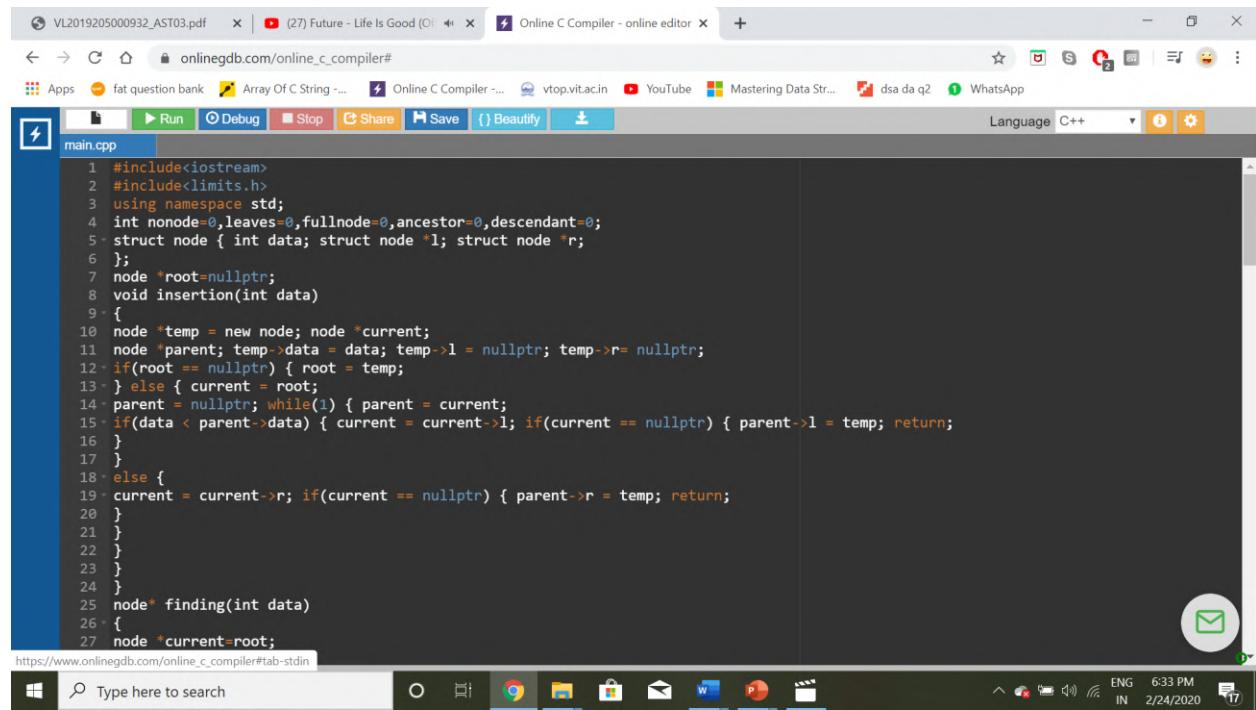
```

Enter the number of edges : 8
Input the nodes of the binary tree in order
parent-child-left<or>right-
1 2 1
2 4 1
4 7 1
7 9 r
2 5 r
1 3 r
3 6 r
6 8 r

Inorder Traversal : 7 9 4 2 5 1 3 6 8
Preorder Traversal : 1 2 4 7 9 5 3 6 8
Postorder Traversal : 9 7 4 5 2 8 6 3 1
Process returned 0 <0x0>   execution time : 24.007 s
Press any key to continue.

```

Question 3



The screenshot shows an online C compiler interface with the following details:

- Title Bar:** VL2019205000932_AST03.pdf | (27) Future - Life Is Good | Online C Compiler - online editor
- Toolbar:** Run, Debug, Stop, Share, Save, Beautify
- Language:** C++
- Code Area (main.cpp):**

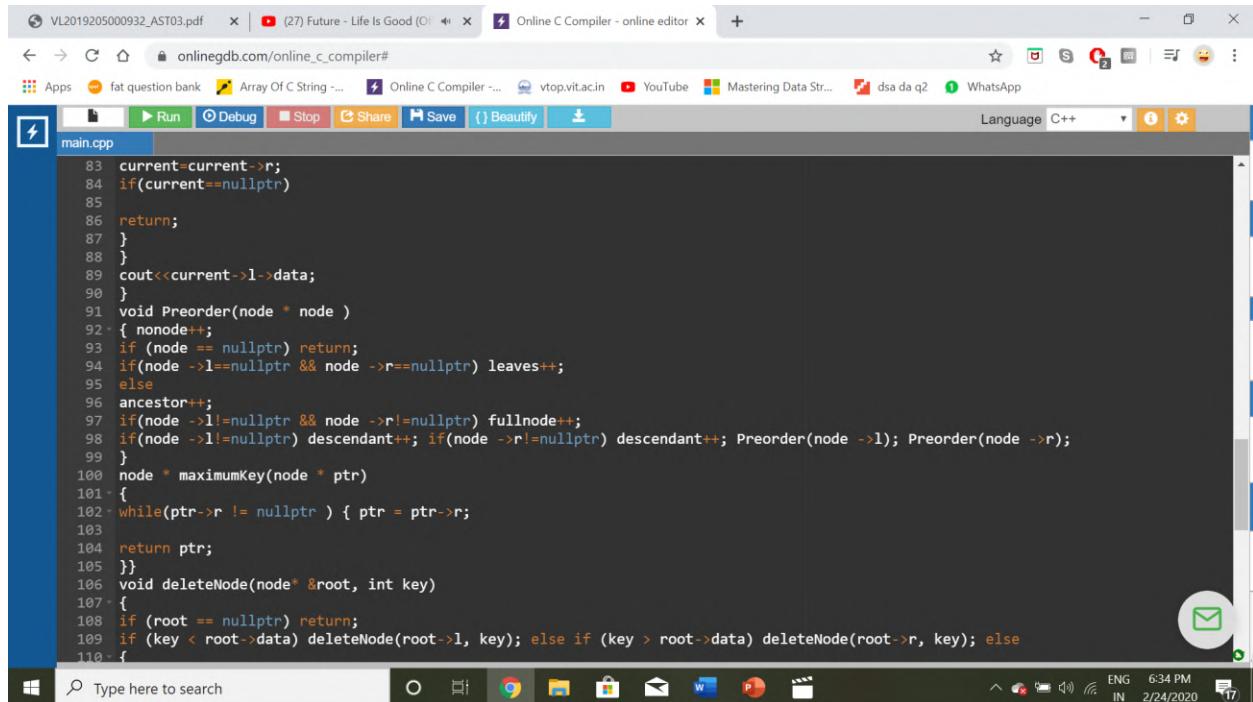
```

1 #include<iostream>
2 #include<limits.h>
3 using namespace std;
4 int nonode=0,leaves=0,fullnode=0,ancestor=0,descendant=0;
5 struct node { int data; struct node *l; struct node *r;
6 };
7 node *root=nullptr;
8 void insertion(int data)
9 {
10    node *temp = new node; node *current;
11    node *parent; temp->data = data; temp->l = nullptr; temp->r= nullptr;
12    if(root == nullptr) { root = temp;
13    } else { current = root;
14    parent = nullptr; while(1) { parent = current;
15    if(data < parent->data) { current = current->l; if(current == nullptr) { parent->l = temp; return;
16    }
17    }
18    else {
19    current = current->r; if(current == nullptr) { parent->r = temp; return;
20    }
21    }
22    }
23    }
24    }
25 node* finding(int data)
26 {
27    node *current=root;

```
- Status Bar:** https://www.onlinegdb.com/online_c_compiler#tab-stdin
- System Tray:** ENG 6:33 PM, IN 2/24/2020, battery icon, signal strength, etc.

```
29 cout<<"Going element after element..."<<endl;
30 while(current->data!=data)
31 {
32     if (current!=nullptr)
33     {
34         cout<<current->data<<" "; if(current->data>data) current=current->l;
35         else current=curr->r; if(current==nullptr) return nullptr;
36     }
37 }
38 return current;
39 }
40 void minelement()
41 {
42     int mi=INT_MAX; node *current=root;
43     while(current!=nullptr)
44     {
45         if(current->data<mi) mi=current->data; current=current->l;
46     }
47 cout<<"\n"<<mi<<endl;
48 }
49 void maxelement()
50 {
51     int ma=INT_MIN; node *current=root;
52     while(current!=nullptr)
53     {
54         if(current->data>ma) ma=current->data; current=current->r;
55     }
56 cout<<"\n"<<ma<<endl;
57 }
```

```
56 cout<<"\n"<<ma<<endl;
57 }
58 void rchild(int data)
59 {
60     node *current=root;
61     cout<<"Going element after element..."<<endl;
62     while(current->data!=data)
63     {
64         if (current!=nullptr)
65         {
66             cout<<current->data<<" "; if(current->data>data) current=current->l;
67             else
68                 current=curr->r; if(current==nullptr) return;
69         }
70     }
71     cout<<current->r->data;
72 }
73 void lchild(int data)
74 {
75     node *current=root;
76     cout<<"Going element after element..."<<endl;
77     while(current->data!=data)
78     {
79         if (current!=nullptr)
80         {
81             cout<<current->data<<" "; if(current->data>data) current=current->l;
82             else
83                 current=curr->r;
```

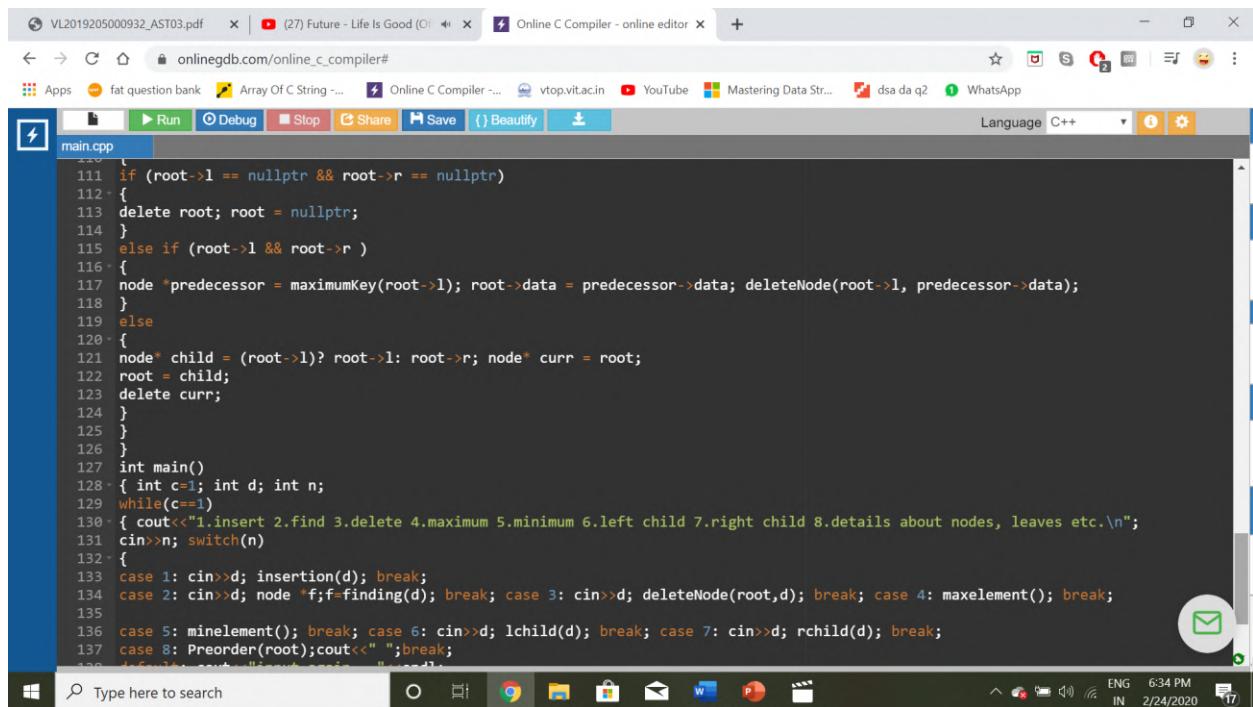


VL2019205000932_AST03.pdf | (27) Future - Life Is Good (O) | Online C Compiler - online editor

onlinegdb.com/online_c_compiler#

main.cpp

```
83     current=current->r;
84     if(current==nullptr)
85     {
86         return;
87     }
88 }
89 cout<<current->l->data;
90 }
91 void Preorder(node * node )
92 {
93     if (node == nullptr) return;
94     if(node ->l==nullptr && node ->r==nullptr) leaves++;
95     else
96         ancestor++;
97     if(node ->l!=nullptr && node ->r!=nullptr) fullnode++;
98     if(node ->l!=nullptr) descendant++; if(node ->r!=nullptr) descendant++; Preorder(node ->l); Preorder(node ->r);
99 }
100 node * maximumKey(node * ptr)
101 {
102     while(ptr->r != nullptr ) { ptr = ptr->r;
103
104     }
105 }
106 void deleteNode(node* &root, int key)
107 {
108     if (root == nullptr) return;
109     if (key < root->data) deleteNode(root->l, key); else if (key > root->data) deleteNode(root->r, key); else
110     {
```

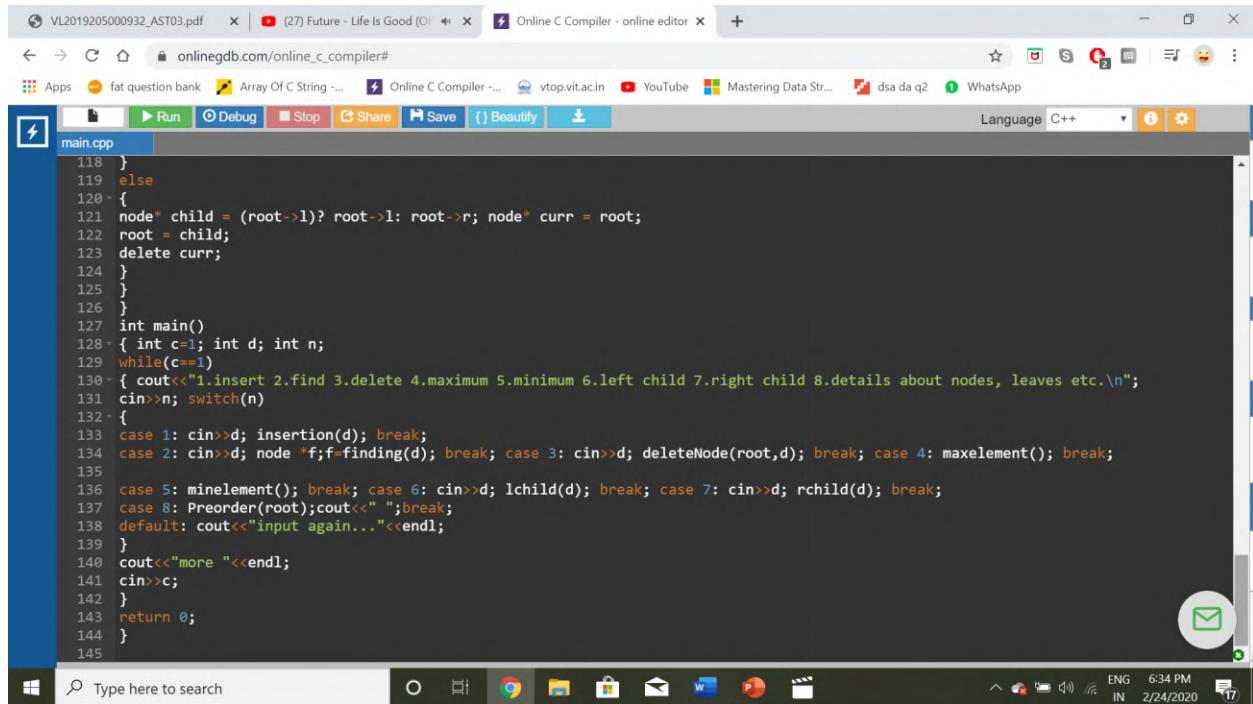


VL2019205000932_AST03.pdf | (27) Future - Life Is Good (O) | Online C Compiler - online editor

onlinegdb.com/online_c_compiler#

main.cpp

```
111     if (root->l == nullptr & root->r == nullptr)
112     {
113         delete root; root = nullptr;
114     }
115     else if (root->l && root->r )
116     {
117         node *predecessor = maximumKey(root->l); root->data = predecessor->data; deleteNode(root->l, predecessor->data);
118     }
119     else
120     {
121         node* child = (root->l)? root->l: root->r; node* curr = root;
122         root = child;
123         delete curr;
124     }
125 }
126 }
127 int main()
128 {
129     int c=1; int d; int n;
130     while(c==1)
131     { cout<<"1.insert 2.find 3.delete 4.maximum 5.minimum 6.left child 7.right child 8.details about nodes, leaves etc.\n";
132     cin>>n; switch(n)
133     {
134         case 1: cin>>d; insertion(d); break;
135         case 2: cin>>d; node *f=f-finding(d); break; case 3: cin>>d; deleteNode(root,d); break; case 4: maxelement(); break;
136         case 5: minelement(); break; case 6: cin>>d; lchild(d); break; case 7: cin>>d; rchild(d); break;
137         case 8: Preorder(root);cout<<"\n";break;
138     }
139 }
```

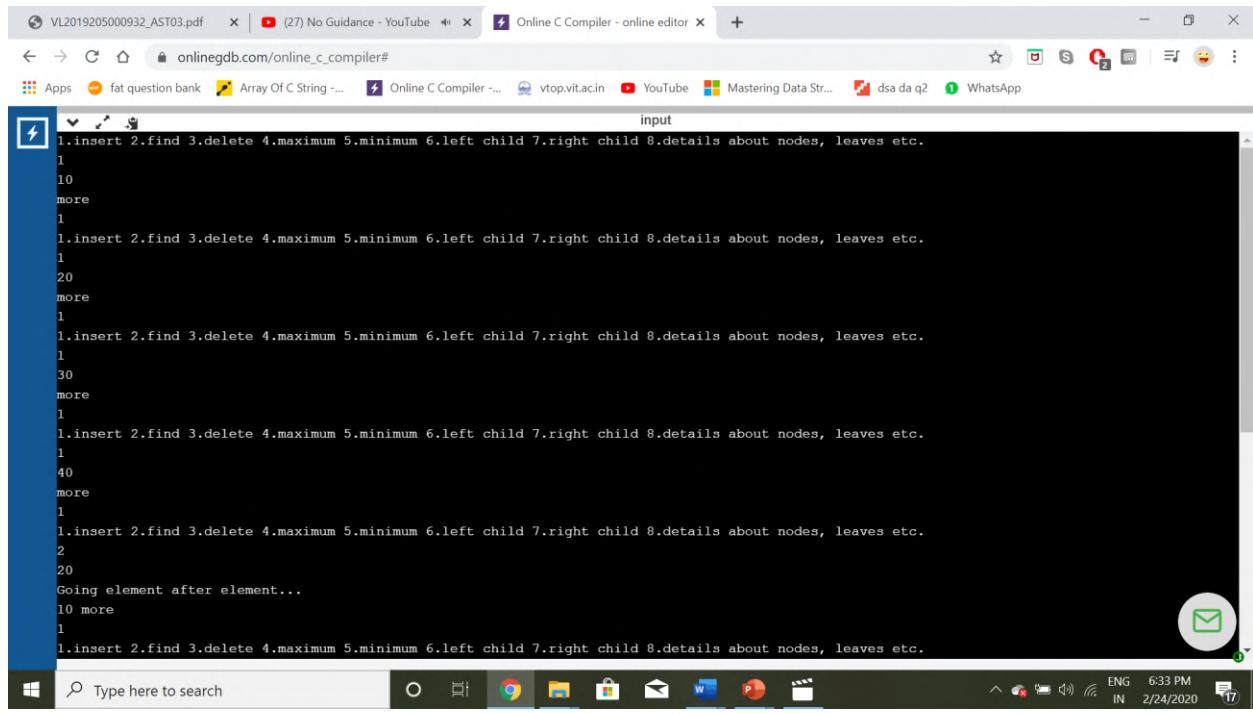


VL2019205000932_AST03.pdf | (27) Future - Life Is Good (O | Online C Compiler - online editor

onlinegdb.com/online_c_compiler#

main.cpp

```
118 }
119 else
120 {
121 node* child = (root->l)? root->l: root->r; node* curr = root;
122 root = child;
123 delete curr;
124 }
125 }
126 }
127 int main()
128 { int c=1; int d; int n;
129 while(c==1)
130 { cout<<"1.insert 2.find 3.delete 4.maximum 5.minimum 6.left child 7.right child 8.details about nodes, leaves etc.\n";
131 cin>n; switch(n)
132 {
133 case 1: cin>d; insertion(d); break;
134 case 2: cin>d; node *f=f-finding(d); break; case 3: cin>d; deleteNode(root,d); break; case 4: maxelement(); break;
135
136 case 5: minelement(); break; case 6: cin>d; lchild(d); break; case 7: cin>d; rchild(d); break;
137 case 8: Preorder(root);cout<<" ";break;
138 default: cout<<"input again..."\<<endl;
139 }
140 cout<<"more "\<<endl;
141 cin>c;
142 }
143 return 0;
144 }
145 }
```



VL2019205000932_AST03.pdf | (27) No Guidance - YouTube | Online C Compiler - online editor

onlinegdb.com/online_c_compiler#

input

```
1.insert 2.find 3.delete 4.maximum 5.minimum 6.left child 7.right child 8.details about nodes, leaves etc.
1
10
more
1
1.insert 2.find 3.delete 4.maximum 5.minimum 6.left child 7.right child 8.details about nodes, leaves etc.
1
20
more
1
1.insert 2.find 3.delete 4.maximum 5.minimum 6.left child 7.right child 8.details about nodes, leaves etc.
1
30
more
1
1.insert 2.find 3.delete 4.maximum 5.minimum 6.left child 7.right child 8.details about nodes, leaves etc.
1
40
more
1
1.insert 2.find 3.delete 4.maximum 5.minimum 6.left child 7.right child 8.details about nodes, leaves etc.
2
20
Going element after element...
10 more
1
1.insert 2.find 3.delete 4.maximum 5.minimum 6.left child 7.right child 8.details about nodes, leaves etc.
```

Question 4

```
1 #include<stdio.h>
2 int q[20],top=-1,front=-1,rear=-1,a[20][20],vis[20],stack[20];
3 int delete();
4 void add(int item); void bfs(int s,int n); void dfs(int s,int n); void push(int item); int pop();
5 void main()
6 {
7     int n,i,s,ch,j;
8     char c,dummy;
9     printf("ENTER THE NUMBER VERTICES ");
10    scanf("%d",&n);
11    for(i=1;i<n;i++)
12    {
13        for(j=1;j<n;j++)
14        {
15            printf("ENTER 1 IF %d HAS A NODE WITH %d ELSE 0 ",i,j);
16            scanf("%d",&a[i][j]);
17        }
18    }
19    printf("THE ADJACENCY MATRIX IS\n");
20    for(i=1;i<n;i++)
21    {
22        for(j=1;j<n;j++)
23        {
24            printf(" %d",a[i][j]);
25        }
26        printf("\n");
27    }
28
29 do
30 {
31     for(i=1;i<=n;i++) vis[i]=0; printf("\nMENU"); printf("\n1.B.F.S");
32     printf("\n2.D.F.S"); printf("\nEnter Your Choice");
33     scanf("%c",&ch);
34     printf("ENTER THE SOURCE VERTEX :");
35     scanf("%d",&s);
36
37     switch(ch)
38     {
39         case 1:bfs(s,n);
40         break; case 2:
41         dfs(s,n);
42         break;
43     }
44     printf("DO U WANT TO CONTINUE(Y/N) ? ");
45     scanf("%c",&dummy);
46     scanf("%c",&c);
47     }while((c=='y')||(c=='Y'));
48 }
49 void bfs(int s,int n)
50 {
51     int p,i; add(s);
52     vis[s]=1; p=delete(); if(p!=0) printf("%d",p);
53     while(p!=0)
54     {
55         for(i=1;i<=n;i++)
56         if((a[p][i]!=0)&(vis[i]==0))
57         {
58             add(i);
59             vis[i]=1;
60         }
61     }
62     n_delete();
63 }
```

```
30 {
31     for(i=1;i<=n;i++) vis[i]=0; printf("\nMENU"); printf("\n1.B.F.S");
32     printf("\n2.D.F.S"); printf("\nEnter Your Choice");
33     scanf("%c",&ch);
34     printf("ENTER THE SOURCE VERTEX :");
35     scanf("%d",&s);
36
37     switch(ch)
38     {
39         case 1:bfs(s,n);
40         break; case 2:
41         dfs(s,n);
42         break;
43     }
44     printf("DO U WANT TO CONTINUE(Y/N) ? ");
45     scanf("%c",&dummy);
46     scanf("%c",&c);
47     }while((c=='y')||(c=='Y'));
48 }
49 void bfs(int s,int n)
50 {
51     int p,i; add(s);
52     vis[s]=1; p=delete(); if(p!=0) printf("%d",p);
53     while(p!=0)
54     {
55         for(i=1;i<=n;i++)
56         if((a[p][i]!=0)&(vis[i]==0))
57         {
58             add(i);
59             vis[i]=1;
60         }
61     }
62     n_delete();
63 }
```

VL2019205000932_AST03.pdf | (27) Future - Life Is Good (O) | Online C Compiler - online editor | (2) WhatsApp

onlinegdb.com/online_c_compiler#

main.c

```
59 Vis[1]=1;
60 }
61 p=delete();
62 if(p!=0)
63 printf("%d ",p);
64 }
65 for(i=1;i<=n;i++) if(vis[i]==0)
66 bts(i,n);
67 }
68 void add(int item)
69 {
70 if(rear==19) printf("QUEUE FULL");
71 else
72 if(rear==1)
73 {
74 q[++rear]=item; front++;
75 }
76 else
77 q[++rear]=item;
78 }
79 }
80 int delete()
81 {
82 int k;
83 if((front>rear)|| (front==1)) return(0);
84 else
85 {
86 k=q[front++];
87 return(k);
88 }
89 }
90 }
```

Type here to search

Language: C

6:52 PM
ENG IN 2/24/2020

VL2019205000932_AST03.pdf | (27) Future - Life Is Good (O) | Online C Compiler - online editor | (2) WhatsApp

onlinegdb.com/online_c_compiler#

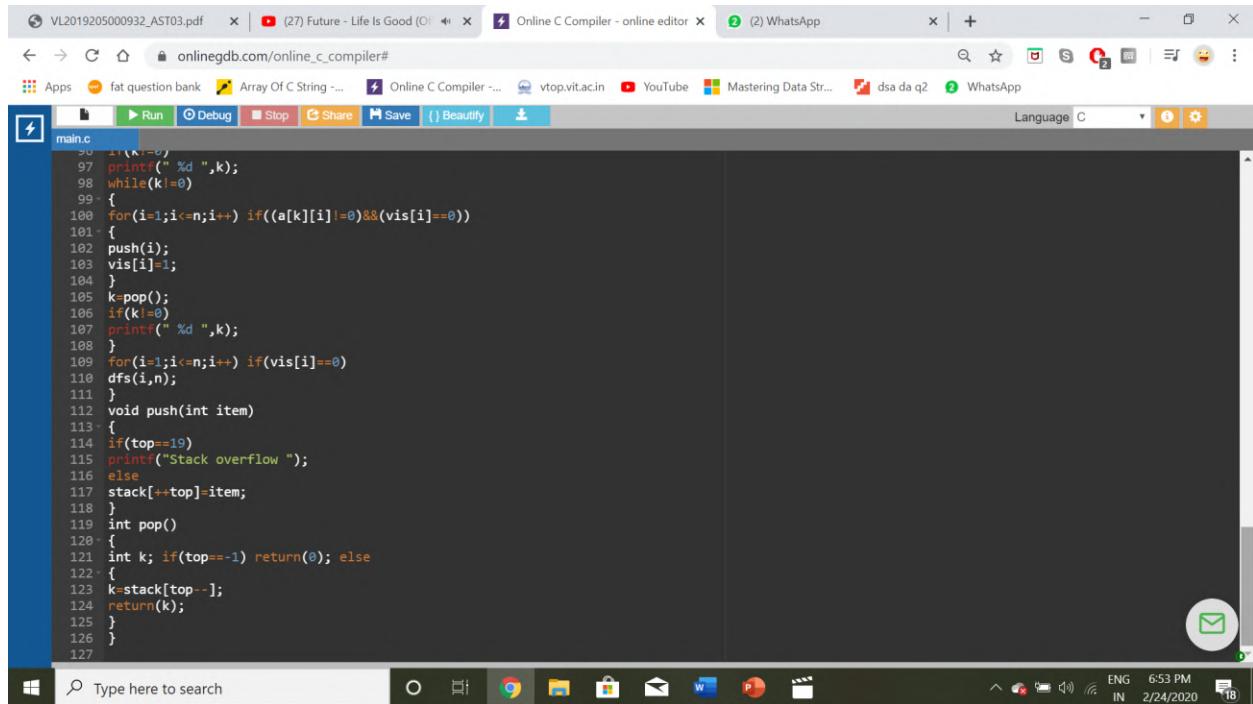
main.c

```
91 void dfs(int s,int n)
92 {
93 int i,k; push(s);
94 vis[s]=1;
95 k=pop();
96 if(k!=0)
97 printf("%d ",k);
98 while(k!=0)
99 {
100 for(i=1;i<=n;i++) if((a[k][i]!=0)&&(vis[i]==0))
101 {
102 push(i);
103 vis[i]=1;
104 }
105 k=pop();
106 if(k!=0)
107 printf("%d ",k);
108 }
109 for(i=1;i<=n;i++) if(vis[i]==0)
110 dfs(i,n);
111 }
112 void push(int item)
113 {
114 if(top==19)
115 printf("Stack overflow ");
116 else
117 stack[++top]=item;
118 }
119 int pop()
120 {
121 int k; if(top==0) return(0); else
122 }
```

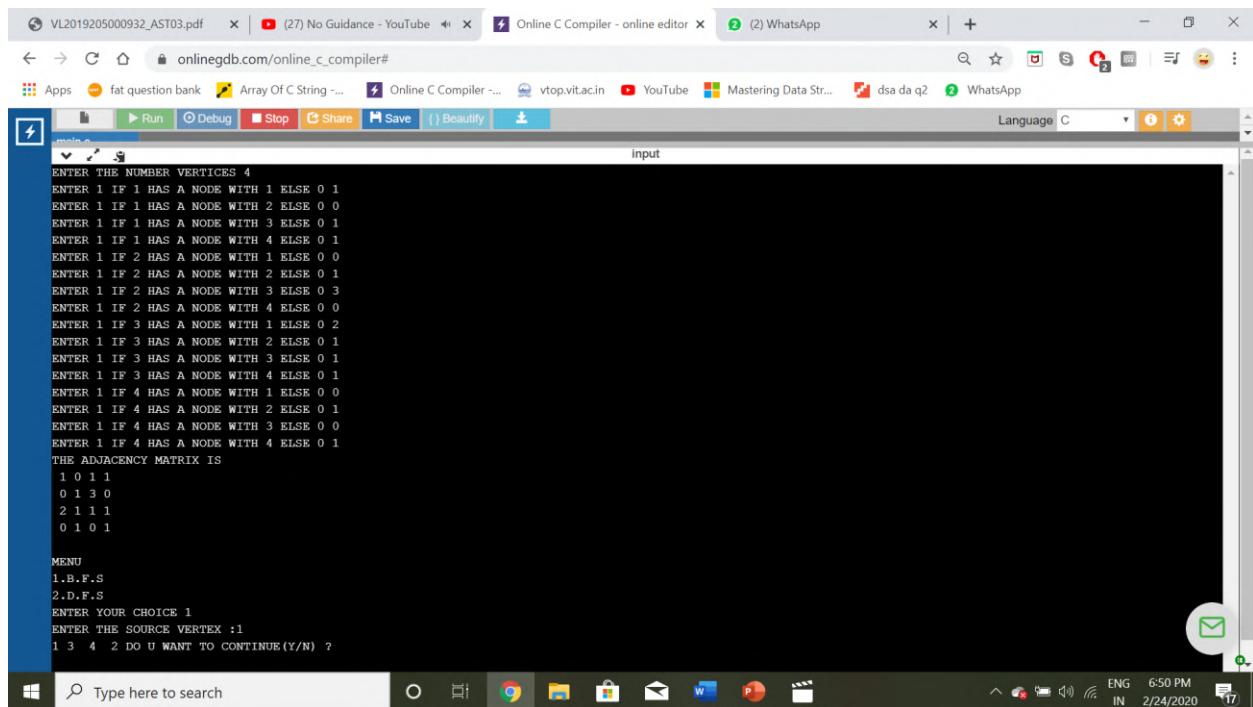
Type here to search

Language: C

6:53 PM
ENG IN 2/24/2020



```
main.c
90    if(k==0)
91    printf("%d ",k);
92    while(k!=0)
93    {
94        for(i=1;i<=n;i++)
95            if((a[k][i]==0)&&(vis[i]==0))
96            {
97                push(i);
98                vis[i]=1;
99            }
100       k=pop();
101       if(k==0)
102           printf("%d ",k);
103       }
104       for(i=1;i<=n;i++)
105           if(vis[i]==0)
106               dfs(i,n);
107       }
108       void push(int item)
109       {
110           if(top==19)
111               printf("Stack overflow ");
112           else
113               stack[++top]=item;
114       }
115       int pop()
116       {
117           int k;
118           if(top== -1) return(0);
119           else
120               k=stack[top--];
121           return(k);
122       }
123   }
124 }
```



```
main.c
ENTER THE NUMBER VERTICES 4
ENTER 1 IF 1 HAS A NODE WITH 1 ELSE 0 1
ENTER 1 IF 1 HAS A NODE WITH 2 ELSE 0 0
ENTER 1 IF 1 HAS A NODE WITH 3 ELSE 0 1
ENTER 1 IF 1 HAS A NODE WITH 4 ELSE 0 1
ENTER 1 IF 2 HAS A NODE WITH 1 ELSE 0 0
ENTER 1 IF 2 HAS A NODE WITH 2 ELSE 0 1
ENTER 1 IF 2 HAS A NODE WITH 3 ELSE 0 3
ENTER 1 IF 2 HAS A NODE WITH 4 ELSE 0 0
ENTER 1 IF 3 HAS A NODE WITH 1 ELSE 0 2
ENTER 1 IF 3 HAS A NODE WITH 2 ELSE 0 1
ENTER 1 IF 3 HAS A NODE WITH 3 ELSE 0 1
ENTER 1 IF 3 HAS A NODE WITH 4 ELSE 0 1
ENTER 1 IF 4 HAS A NODE WITH 1 ELSE 0 0
ENTER 1 IF 4 HAS A NODE WITH 2 ELSE 0 1
ENTER 1 IF 4 HAS A NODE WITH 3 ELSE 0 0
ENTER 1 IF 4 HAS A NODE WITH 4 ELSE 0 1
THE ADJACENCY MATRIX IS
1 0 1 1
0 1 3 0
2 1 1 1
0 1 0 1

MENU
1.B.F.S
2.D.F.S
ENTER YOUR CHOICE 1
ENTER THE SOURCE VERTEX :1
1 3 4 2 DO U WANT TO CONTINUE(Y/N) ?
```

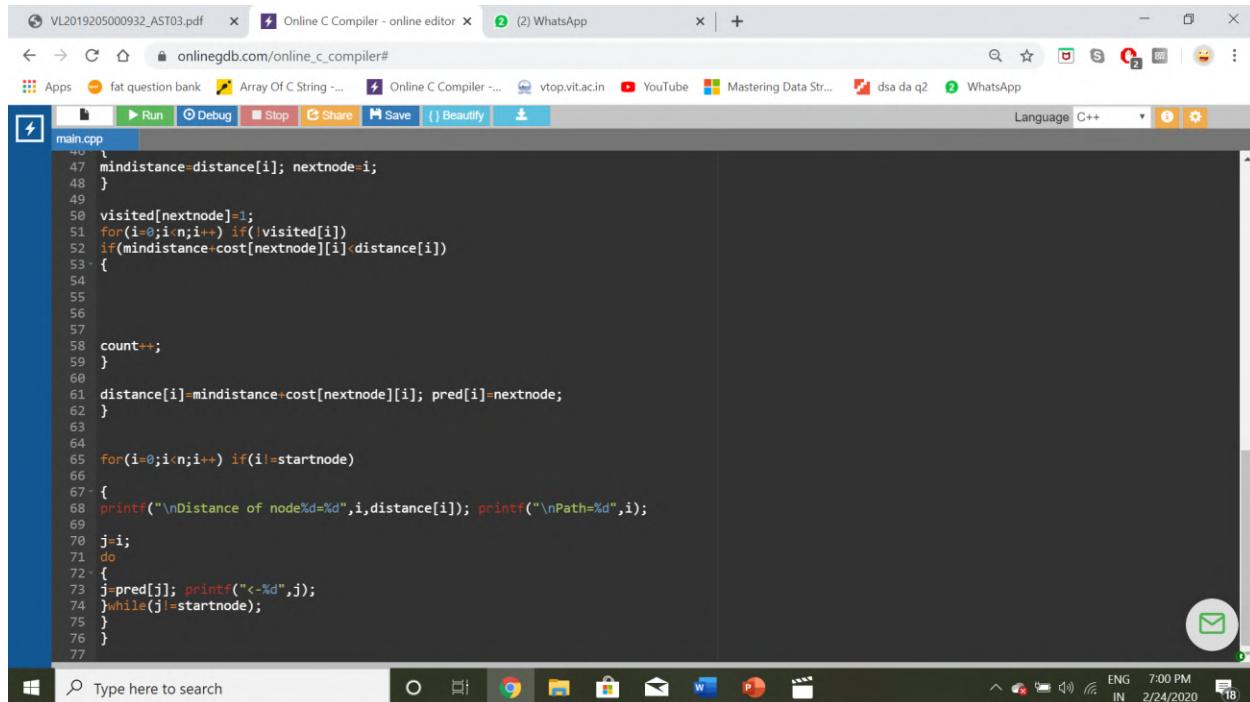
The screenshot shows a Windows desktop environment. At the top, there is a taskbar with several pinned icons: Apps, fat.question bank, Array Of C String ..., Online C Compiler - online editor, (2) WhatsApp, YouTube, vtop.vit.ac.in, Mastering Data Str..., dsa da q2, and WhatsApp. Below the taskbar, a browser window is open with the URL onlinegdb.com/online_c_compiler#. The browser tab bar also lists: VL2019205000932_AST03.pdf, (27) No Guidance - YouTube, Online C Compiler - online editor, and (2) WhatsApp. The main content of the browser shows a C program running in an online compiler. The code is as follows:

```
ENTER THE NUMBER VERTICES 3
ENTER 1 IF 1 HAS A NODE WITH 1 ELSE 0 1
ENTER 1 IF 1 HAS A NODE WITH 2 ELSE 0 1
ENTER 1 IF 1 HAS A NODE WITH 3 ELSE 0 0
ENTER 1 IF 2 HAS A NODE WITH 1 ELSE 0 0
ENTER 1 IF 2 HAS A NODE WITH 2 ELSE 0 1
ENTER 1 IF 2 HAS A NODE WITH 3 ELSE 0 1
ENTER 1 IF 3 HAS A NODE WITH 1 ELSE 0 0
ENTER 1 IF 3 HAS A NODE WITH 2 ELSE 0 1
ENTER 1 IF 3 HAS A NODE WITH 3 ELSE 0 0
THE ADJACENCY MATRIX IS
1 1 0
0 1 1
0 1 0

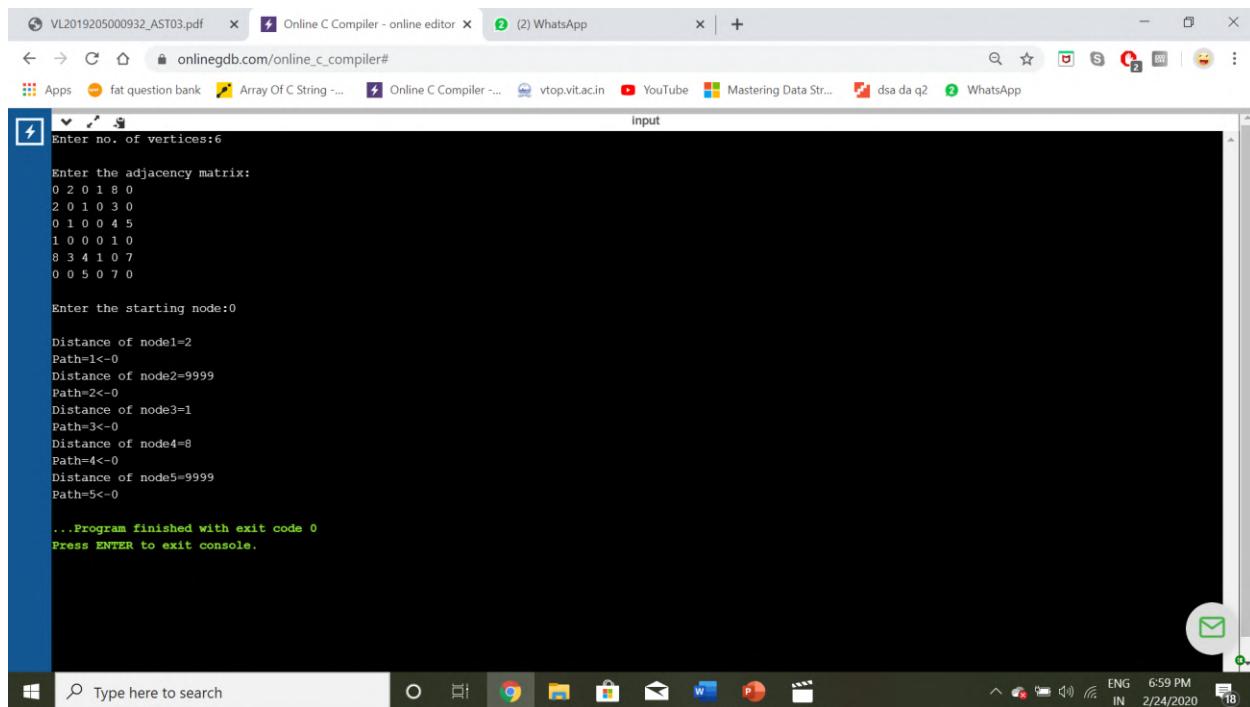
MENU
1.B.F.S
2.D.F.S
ENTER YOUR CHOICE2
ENTER THE SOURCE VERTEX :1
1 2 3 DO U WANT TO CONTINUE(Y/N) ?
```

Question 5

```
1 #include<stdio.h>
2 #include<conio.h>
3 #define INFINITY 9999
4 #define MAX 10
5
6 void dijkstra(int G[MAX][MAX],int n,int startnode);
7
8 int main()
9 {
10    int G[MAX][MAX],i,j,n,u; printf("Enter no. of vertices:"); scanf("%d",&n);
11    printf("\nEnter the adjacency matrix:\n");
12
13    for(i=0;i<n;i++) for(j=0;j<n;j++)
14        scanf("%d",&G[i][j]);
15
16    printf("\nEnter the starting node:"); scanf("%d",&u);
17    dijkstra(G,n,u);
18
19    return 0;
20 }
21
22 void dijkstra(int G[MAX][MAX],int n,int startnode)
23 {
24
25    int cost[MAX][MAX],distance[MAX],pred[MAX];
26    int visited[MAX],count,mindistance,nextnode,i,j;
27
28    for(i=0;i<n;i++) for(j=0;j<n;j++)
29        if(G[i][j]==0) cost[i][j]=INFINITY;
30    else
31        cost[i][j]=G[i][j];
32
33    for(i=0;i<n;i++)
34    {
35        distance[i]=cost[startnode][i]; pred[i]=startnode; visited[i]=0;
36    }
37
38    distance[startnode]=0; visited[startnode]=1; count=1;
39
40    while(count<n-1)
41    {
42        mindistance=INFINITY;
43
44        for(i=0;i<n;i++) if(distance[i]<mindistance&&!visited[i])
45        {
46            mindistance=distance[i]; nextnode=i;
47        }
48
49        visited[nextnode]=1;
50        for(i=0;i<n;i++) if(!visited[i])
51            if(mindistance+cost[nextnode][i]<distance[i])
52            {
53
54
55
56            count++;
57        }
58        distance[i]=mindistance+cost[nextnode][i]; pred[i]=nextnode;
59
60    }
61 }
```



```
main.cpp
+-
47 mindistance=distance[i]; nextnode=i;
48 }
49
50 visited[nextnode]=1;
51 for(i=0;i<n;i++) if(!visited[i])
52 if(mindistance>cost[nextnode][i]>distance[i])
53 {
54
55
56
57 count++;
58 }
59
60 distance[i]=mindistance+cost[nextnode][i]; pred[i]=nextnode;
61
62
63
64
65 for(i=0;i<n;i++) if(i!=startnode)
66 {
67
68 printf("\nDistance of node%d=%d",i,distance[i]); printf("\nPath=%d",i);
69
70 j=i;
71 do
72 {
73 j=pred[j]; printf("<-%d",j);
74 }while(j!=startnode);
75
76 }
77
```



```
VL2019205000932_AST03.pdf
Online C Compiler - online editor
(2) WhatsApp
Type here to search
O
C
H
onlinegdb.com/online_c_compiler
Apps fat question bank Array Of C String ... Online C Compiler ... vtop.vit.ac.in YouTube Mastering Data Str... dsa da q2 WhatsApp
Language C++
File Run Debug Stop Share Save Beautify
main.cpp
Enter no. of vertices:6
Enter the adjacency matrix:
0 2 0 1 8 0
2 0 1 0 3 0
0 1 0 0 4 5
1 0 0 0 1 0
8 3 4 1 0 7
0 0 5 0 7 0
Enter the starting node:0
Distance of node1=0
Path=1<-0
Distance of node2=9999
Path=2<-0
Distance of node3=1
Path=3<-0
Distance of node4=8
Path=4<-0
Distance of node5=9999
Path=5<-0
...
Program finished with exit code 0
Press ENTER to exit console.
```

Question 6

```
1 #include<stdio.h>
2 #include<conio.h>
3 #include<stdlib.h>
4 int i,j,k,a,b,u,v,n,ne=1;
5 int min,mincost=0,cost[9][9],parent[9];
6 int find(int); int uni(int,int);
7 int main()
8 {
9     printf("\nEnter the no. of vertices: "); scanf("%d",&n);
10    printf("\nEnter the cost adjacency matrix:\n");
11    for(i=1;i<=n;i++)
12    {
13        for(j=1;j<=n;j++)
14        {
15            scanf("%d",&cost[i][j]);
16            if(cost[i][j]==0)
17                cost[i][j]=999;
18        }
19    }
20    printf("The edges of Minimum Cost Spanning Tree are\n");
21    while(ne < n)
22    {
23        for(i=1,min=999;i<=n;i++)
24        {
25            for(j=1;j <= n;j++)
26            {
27                if(cost[i][j] < min)
28                {
29                    min=cost[i][j]; a=u=i;
30                    b=v=j;
31                }
32            }
33        }
34        u=find(u); v=find(v); if(uni(u,v))
35        {
36            printf("%d edge (%d,%d) =%d\n",ne++,a,b,min); mincost +=min;
37        }
38        cost[a][b]-cost[b][a]=999;
39    }
40    printf("\n\tMinimum cost = %d\n",mincost); getch();
41 }
42 int find(int i)
43 {
44     if(parent[i]==i)
45         return i;
46     else
47         int uni(int i,int j)
48     {
49         if(i==j)
50         {
51             parent[j]=i;
52             return 1;
53         }
54         return 0;
55     }
56 }
```

```
1 #include<stdio.h>
2 #include<conio.h>
3 #include<stdlib.h>
4 int i,j,k,a,b,u,v,n,ne=1;
5 int min,mincost=0,cost[9][9],parent[9];
6 int find(int); int uni(int,int);
7 int main()
8 {
9     printf("\nEnter the no. of vertices: "); scanf("%d",&n);
10    printf("\nEnter the cost adjacency matrix:\n");
11    for(i=1;i<=n;i++)
12    {
13        for(j=1;j<=n;j++)
14        {
15            scanf("%d",&cost[i][j]);
16            if(cost[i][j]==0)
17                cost[i][j]=999;
18        }
19    }
20    printf("The edges of Minimum Cost Spanning Tree are\n");
21    while(ne < n)
22    {
23        for(i=1,min=999;i<=n;i++)
24        {
25            for(j=1;j <= n;j++)
26            {
27                if(cost[i][j] < min)
28                {
29                    min=cost[i][j]; a=u=i;
30                    b=v=j;
31                }
32            }
33        }
34        u=find(u); v=find(v); if(uni(u,v))
35        {
36            printf("%d edge (%d,%d) =%d\n",ne++,a,b,min); mincost +=min;
37        }
38        cost[a][b]-cost[b][a]=999;
39    }
40    printf("\n\tMinimum cost = %d\n",mincost); getch();
41 }
42 int find(int i)
43 {
44     if(parent[i]==i)
45         return i;
46     else
47         int uni(int i,int j)
48     {
49         if(i==j)
50         {
51             parent[j]=i;
52             return 1;
53         }
54         return 0;
55     }
56 }
```

VL201920500932_AST03.pdf

Online C Compiler - online editor

WhatsApp

onlinegdb.com/online_c_compiler#

Apps

fat question bank

Array Of C String ...

Online C Compiler - ...

vtop.vit.ac.in

YouTube

Mastering Data Str...

dsa da q2

WhatsApp

input

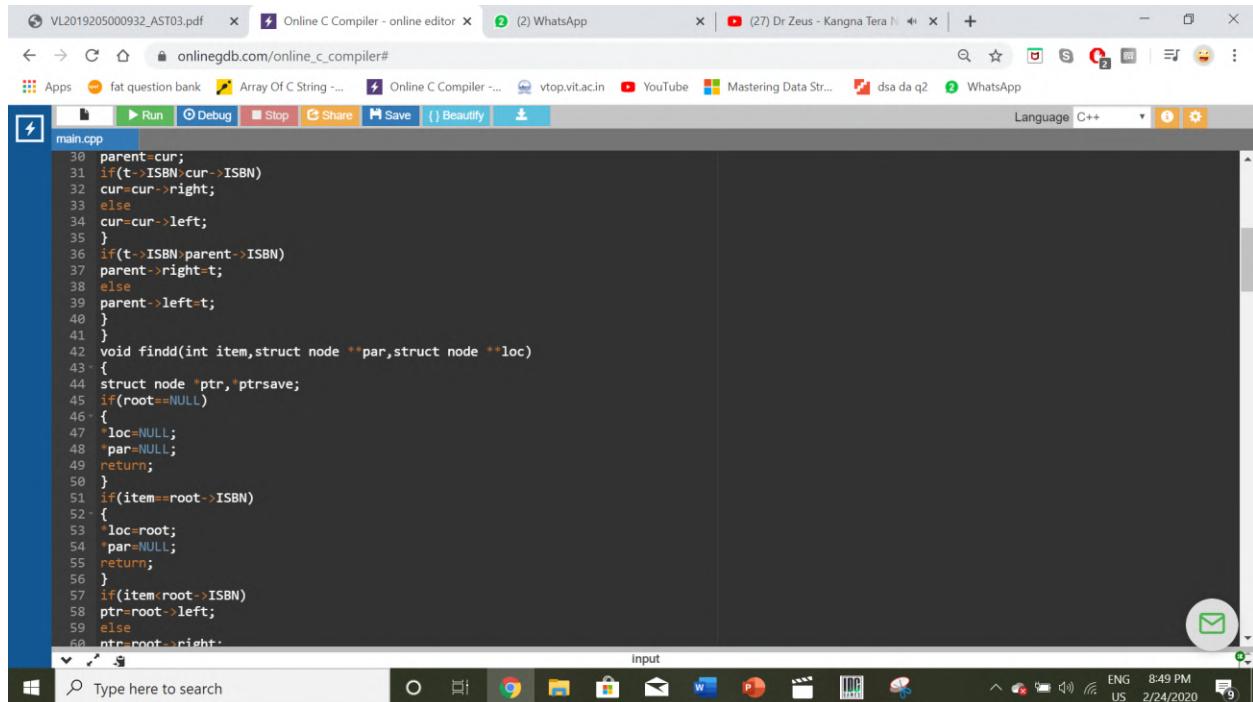
```
Enter the no. of vertices: 4

Enter the cost adjacency matrix:
1 2 3 4
5 6 7 8
9 0 11 12
13 14 15 16
The edges of Minimum Cost Spanning Tree are
1 edge (1,2) =2
2 edge (1,3) =3
3 edge (1,4) =4

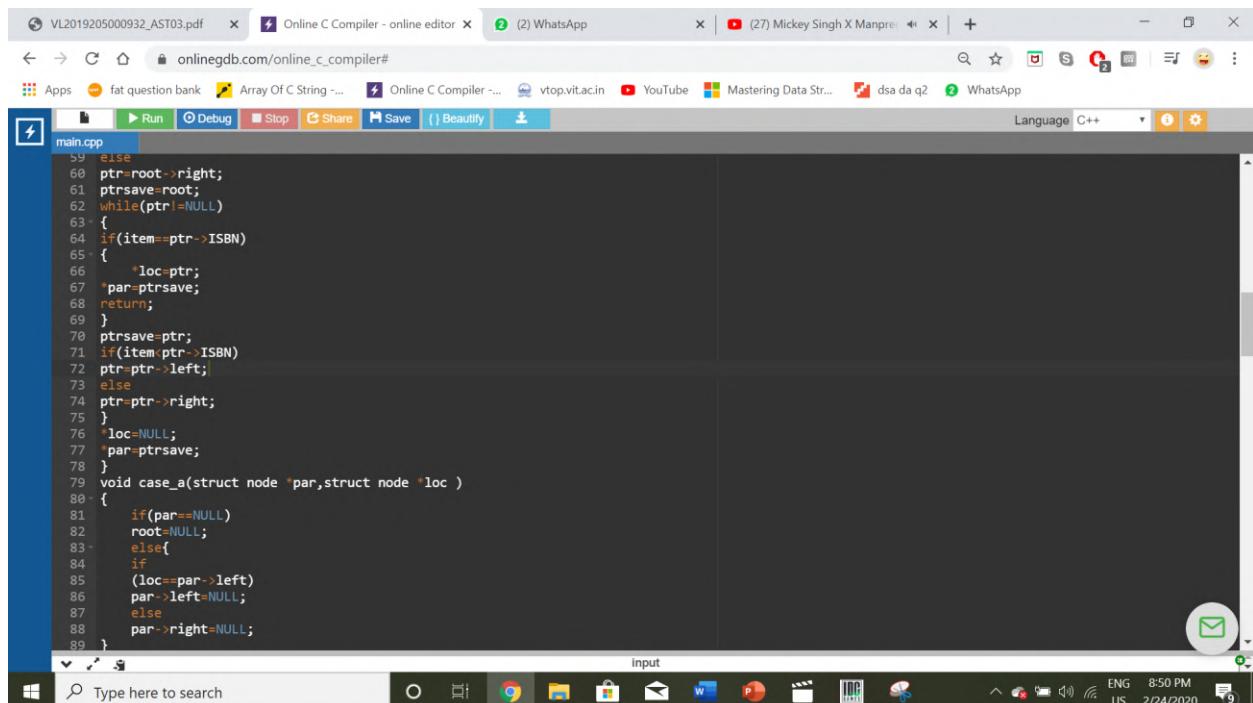
        Minimum cost = 9

...Program finished with exit code 0
Press ENTER to exit console.
```

Question 7



```
30     parent=cur;
31     if(t->ISBN==cur->ISBN)
32         cur=cur->right;
33     else
34         cur=cur->left;
35     }
36     if(t->ISBN==parent->ISBN)
37         parent->right=t;
38     else
39         parent->left=t;
40     }
41 }
42 void findd(int item,struct node **par,struct node **loc)
43 {
44     struct node *ptr,*ptrsave;
45     if(root==NULL)
46     {
47         *loc=NULL;
48         *par=NULL;
49         return;
50     }
51     if(item==root->ISBN)
52     {
53         *loc=root;
54         *par=NULL;
55         return;
56     }
57     if(item<root->ISBN)
58         ptr=root->left;
59     else
60         ptr=root->right;
```



```
59     else
60         ptr=root->right;
61     ptrsave=root;
62     while(ptr!=NULL)
63     {
64         if(item==ptr->ISBN)
65         {
66             *loc=ptr;
67             *par=ptrsave;
68             return;
69         }
70         ptrsave=ptr;
71         if(item<ptr->ISBN)
72             ptr=ptr->left;
73         else
74             ptr=ptr->right;
75     }
76     *loc=NULL;
77     *par=ptrsave;
78 }
79 void case_a(struct node *par,struct node *loc )
80 {
81     if(par==NULL)
82         root=NULL;
83     else{
84         if(
85             (loc==par->left)
86             par->left=NULL;
87         else
88             par->right=NULL;
89     }
```

VL2019205000932_AST03.pdf

Online C Compiler - online editor

(2) WhatsApp

onlinegdb.com/online_c_compiler

main.cpp

```
88     par->right=NULL;
89 }
90 }
91 void case_b(struct node *par,struct node *loc)
92 {
93     struct node *child;
94     if(loc->left!=NULL)
95         child=loc->left;
96     else
97         child=loc->right;
98     if(par==NULL)
99         root=child;
100    else
101    {
102        if( loc==par->left)
103            par->left=child;
104        else
105            par->right=child;
106    }
107 }
108 void case_c(struct node *par,struct node *loc)
109 {
110     struct node *ptr,*ptrsave,*suc,*parsuc; ptrsave=loc;
111     ptr=loc->right;
112     while(ptr->left!=NULL)
113     {
114         ptrsave=ptr;
115         ptr=ptr->left;
116     }
117     suc=ptr; parsuc=ptrsave;
118     if(suc->left==NULL && suc->right==NULL)
```

VL2019205000932_AST03.pdf

Online C Compiler - online editor

(2) WhatsApp

onlinegdb.com/online_c_compiler

main.cpp

```
117     suc=parsuc,suc;
118     if(suc->left==NULL && suc->right==NULL)
119         case_a(parsuc,suc);
120     else
121         case_b(parsuc,suc);
122     if(par==NULL)
123         root=suc;
124     else
125     {
126         if(loc==par->left)
127             par->left=suc;
128         else
129             par->right=suc;
130         suc->left=loc->left;
131         suc->right=loc->right;
132     }
133 }
134 int del(int item)
135 {
136     struct node *parent,*location;
137     if(root==NULL)
138     {
139         cout<<"Tree empty";
140         return 0;
141     }
142     findd(item,&parent,&location);
143     if(location==NULL)
144     {
145         cout<<"Item not present in tree";
146         return 0;
147     }
```

```
main.cpp
147 }
148 if(location->left==NULL && location->right==NULL)
149 case_a(parent,location);
150 if(location->left!=NULL && location->right==NULL)
151 case_b(parent,location);
152 if(location->left==NULL && location->right!=NULL)
153 case_b(parent,location);
154 if(location->left!=NULL && location->right!=NULL)
155 case_c(parent,location);
156 free(location);
157 }
158 void searc(int d,char x)
159 {
160 struct node cur,*parent; cur=root;
161 parent=root;
162 while(cur->ISBN!=d)
163 {
164 parent=cur;
165 if(d<cur->ISBN)
166 cur=cur->right;
167 else
168 cur=cur->left;
169 }
170 if(x=='r')
171 strcpy(cur->status,"on-shelf");
172 else if(x=='b')
173 strcpy(cur->status,"borrowed");
174 }
175 void preorder(struct node*t)
176 {
```

```
main.cpp
186 char c,a[20],t[20];
187 while(c!=e')
188 {
189 cout<<"\nEnter a for ADD BOOK\n b for BORROW BOOK\n r for RETURN BOOK\n d for REMOVE BOOK\n I for PRINT BOOKS\n e. EXIT\n";
190 cin>>c;
191 switch(c)
192 {
193     case 'a':
194         cout<<"Enter ISBN: ";
195         cin>>isbn;
196         cout<<"Enter book author: ";
197         cin>>a;
198         cout<<"Enter book title: ";
199         cin>>t; ins(isbn,a,t); break;
200     case 'b':cout<<"\nEnter ISBN to borrow: ";
201         cin>>isbn; searc(isbn,'b'); break;
202     case 'r':
203         cout<<"\nEnter ISBN to return: ";
204         cin>>isbn; searc(isbn,'r'); break;
205     case 'd':
206         cout<<"\nEnter ISBN to remove: ";
207         cin>>isbn; del(isbn); break;
208     case 'I':
209         preorder(root);
210         break;
211     }
212 }
213 return 0;
214 }
215 }
```

```
Enter a for ADD BOOK
b for BORROW BOOK
r for RETURN BOOK
d for REMOVE BOOK
I for PRINT BOOKS
e. EXIT
a
Enter ISBN: 1234567
Enter book author: Kulvir Singh
Enter book title:
Enter a for ADD BOOK
b for BORROW BOOK
r for RETURN BOOK
d for REMOVE BOOK
I for PRINT BOOKS
e. EXIT
a
Enter ISBN: 1234599
Enter book author: geroge
Enter book title: hello world
```

```
Enter a for ADD BOOK
b for BORROW BOOK
r for RETURN BOOK
d for REMOVE BOOK
I for PRINT BOOKS
e. EXIT
a
Enter ISBN: 1234
Enter book author: hello
Enter book title: hello

Enter a for ADD BOOK
b for BORROW BOOK
r for RETURN BOOK
d for REMOVE BOOK
I for PRINT BOOKS
e. EXIT
b

Enter ISBN to borrow: 1234
```

```
d for REMOVE BOOK
I for PRINT BOOKS
e. EXIT
a
Enter ISBN: 1234
Enter book author: kulvir
Enter book title: hello

Enter a for ADD BOOK
b for BORROW BOOK
r for RETURN BOOK
d for REMOVE BOOK
I for PRINT BOOKS
e. EXIT
I
1234 kulvir hello on-shelf
```

VL2019205000932_AST03.pdf Online C Compiler - online editor (2) WhatsApp (27) Pyar Bolda (feat. Gur Sid input

onlinegdb.com/online_c_compiler#

Apps fat.question bank Array Of C String ... Online C Compiler ... vtop.vit.ac.in YouTube Mastering Data Str... dsa da q2 WhatsApp

I For PRINT BOOKS
e. EXIT
a
Enter ISBN: 1234
Enter book author: kulvir
Enter book title: hello

Enter a for ADD BOOK
b for BORROW BOOK
r for RETURN BOOK
d for REMOVE BOOK
I for PRINT BOOKS
e. EXIT
d

Enter ISBN to remove: 1234

Enter a for ADD BOOK
b for BORROW BOOK
r for RETURN BOOK
d for REMOVE BOOK
I for PRINT BOOKS
e. EXIT
I

Enter a for ADD BOOK
b for BORROW BOOK
r for RETURN BOOK
d for REMOVE BOOK
I for PRINT BOOKS
e. EXIT

Type here to search

8:55 PM
ENG 2/24/2020 9