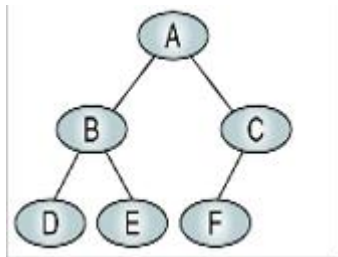


Ripunjay Narula (19BCE0470)
Assignment-7

Implement the binary tree using an array.



Code:

```
#include <stdio.h>
#include <stdlib.h>
char tree[10];
void root(char key)
{
    if(tree[0]!='\0')
        printf("Root already present");
    else
        tree[0]=key;
}
void lchild(char key,int parent)
{
    if(tree[parent]=='\0')
        printf( "\nCan't put child at %d, parent not present ", ((parent*2)+ 1));
    else
        tree[(parent*2)+ 1]=key;
}
void rchild(char key,int parent)
{
    if(tree[parent]=='\0')
        printf( "\nCan't put child at %d, parent not present ", ((parent*2)+ 2));
    else
        tree[(parent*2)+ 2]=key;
}
void ip_tree()
```

```

{
int i;
printf("\n");
for(i=0;i<10;i++)
{
if(tree[i] != '\0')
printf("%c ",tree[i]);
else
printf(" ");
}
}
int main()
{
root('A');
lchild('B',0);
rchild('C',0);
lchild('D',1);
rchild('E',1);
lchild('F',2);
ip_tree();
return 0;
}

```

Screenshot:

The screenshot shows the OnlineGDB website interface. The code editor displays the following C++ code:

```

21 printf("Can't put child at %d, parent not present", ((parent*2)+2));
22 else
23 tree[(parent*2)+2]=key;
24 }
25 void ip_tree()
26 {
27 int i;
28 printf("\n");
29 for(i=0;i<10;i++)
30 {
31 if(tree[i] != '\0')
32 printf("%c ",tree[i]);
33 else
34 printf(" ");
35 }
36 }
37 int main()
38 {
39 root('A');
40 lchild('B',0);
41 rchild('C',0);
42 lchild('D',1);
43 rchild('E',1);
44 lchild('F',2);
45 ip_tree();
46 return 0;
47 }

```

The output console shows the following text:

```

A B C D E F
...Program finished with exit code 0
Press ENTER to exit console.

```

The website interface includes a sidebar with navigation links such as "code, compile, run, debug, share", "IDE", "My Projects", "Learn Programming", "Programming Questions", "Jobs", "Sign Up", and "Login". The footer contains links for "About", "FAQ", "Blog", "Terms of Use", "Contact Us", "GDB", "Tutorial", "Credits", and "Privacy".