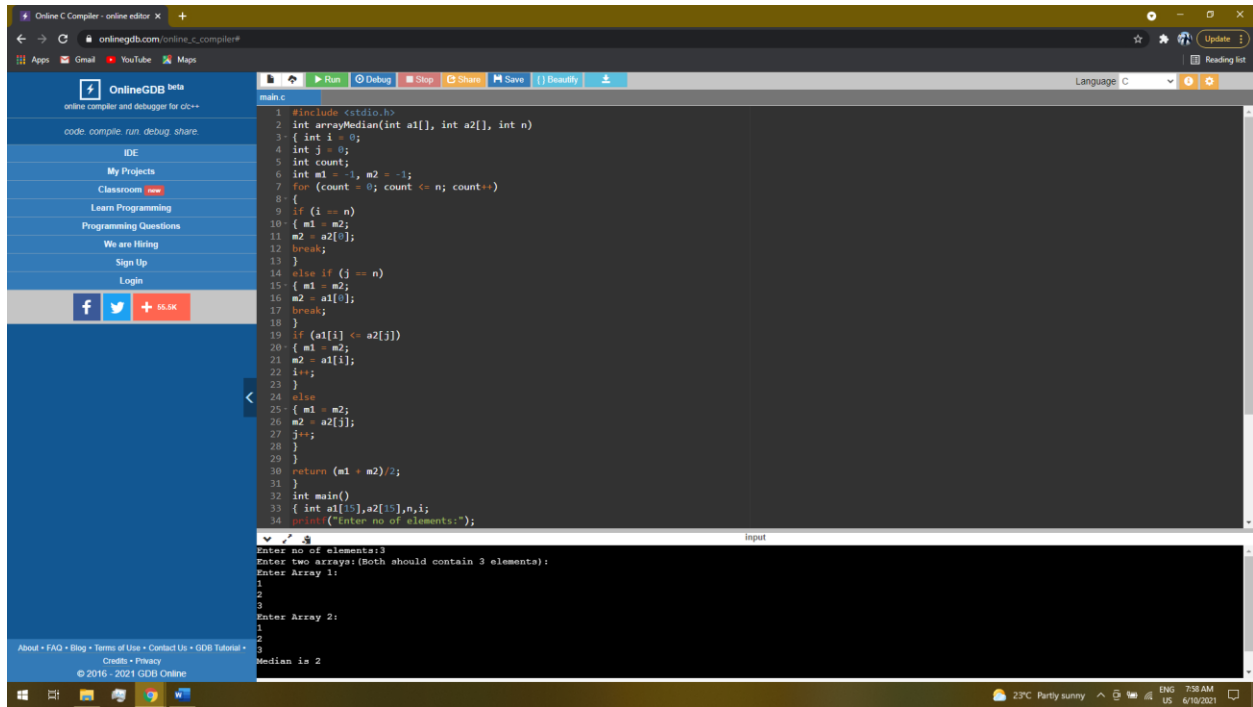


OUTPUT FOR MEDIAN



The screenshot displays the OnlineGDB web interface. On the left is a sidebar with navigation links like 'IDE', 'My Projects', 'Classroom', and 'Learn Programming'. The main area shows a C program for finding the median of two arrays. The code includes a function `arrayMedian` that iterates through two arrays, `a1` and `a2`, to find the median element. The `main` function prompts the user to enter the number of elements and the two arrays. The output at the bottom shows the user input: 3 elements, array 1: 1 2 3, array 2: 1 2 3, and the resulting median value of 2.

```
1 #include <stdio.h>
2 int arrayMedian(int a1[], int a2[], int n)
3 { int i = 0;
4   int j = 0;
5   int count;
6   int m1 = -1, m2 = -1;
7   for (count = 0; count <= n; count++)
8   {
9     if (i == n)
10    { m1 = a2[j];
11      break;
12    }
13    else if (j == n)
14    { m2 = a1[i];
15      break;
16    }
17    if (a1[i] <= a2[j])
18    { m1 = a2[j];
19      m2 = a1[i];
20      i++;
21    }
22    else
23    { m1 = m2;
24      m2 = a2[j];
25      j++;
26    }
27  }
28  return (m1 + m2)/2;
29 }
30
31 int main()
32 { int a1[15], a2[15], n, i;
33   printf("Enter no. of elements:");
34   int n;
35   scanf("%d", &n);
36   printf("Enter two arrays: (Both should contain %d elements):", n);
37   for (i = 0; i < n; i++)
38   { scanf("%d", &a1[i]);
39     scanf("%d", &a2[i]);
40   }
41   int median = arrayMedian(a1, a2, n);
42   printf("Median is %d", median);
43 }
```

Input

```
Enter no. of elements:3
Enter two arrays: (Both should contain 3 elements):
1
2
3
Enter Array 2:
1
2
3
Median is 2
```

The screenshot shows the OnlineGDB website interface. On the left is a sidebar with navigation links like 'IDE', 'My Projects', 'Classrooms', 'Learn Programming', 'Programming Questions', 'We are Hiring', 'Sign Up', and 'Login'. The main area is divided into three sections: a code editor, an input/output console, and a terminal. The code editor contains a C program that finds the median of two arrays. The input console shows the user entering '3' for the number of elements, followed by two arrays: [1, 2, 3] and [1, 2, 3]. The output console shows the program's execution, including prompts for array elements and the final result 'Median is 2'. The terminal at the bottom shows the program finishing with exit code 0.

```
13 }
14 else if (j == n)
15 { m1 = m2;
16 m2 = a1[j];
17 break;
18 }
19 if (a1[i] <= a2[j])
20 { m1 = m2;
21 m2 = a1[i];
22 i++;
23 }
24 else
25 { m1 = m2;
26 m2 = a2[j];
27 j++;
28 }
29 }
30 return (m1 + m2)/2;
31 }
32 int main()
33 { int a1[15], a2[15], n, i;
34 printf("Enter no of elements:");
35 scanf("%d", &n);
36 printf("Enter two arrays:(Both should contain %d elements):\n", n);
37 printf("Enter Array 1:\n");
38 for(i=0; i<n; i++)
39 scanf("%d", &a1[i]);
40 printf("Enter Array 2:\n");
41 for(i=0; i<n; i++)
42 scanf("%d", &a2[i]);
43 printf("Median is %d", arrayMedian(a1, a2, n));
44 getch();
45 return 0;
46 }
```

Input

```
Enter no of elements:3
Enter two arrays:(Both should contain 3 elements):
Enter Array 1:
1
2
3
Enter Array 2:
1
2
3
Median is 2
```

Output

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

CODE FOR MEDIAN

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

```
int arrayMedian(int a1[], int a2[], int n)
```

```
{ int i = 0;
```

```
int j = 0;
```

```

int count;

int m1 = -1, m2 = -1;

for (count = 0; count <= n; count++)
{
    if (i == n)
    { m1 = m2;
      m2 = a2[0];
      break;
    }
    else if (j == n)
    { m1 = m2;
      m2 = a1[0];
      break;
    }
    if (a1[i] <= a2[j])
    { m1 = m2;
      m2 = a1[i];
      i++;
    }
    else
    { m1 = m2;
      m2 = a2[j];
      j++;
    }
}

return (m1 + m2)/2;

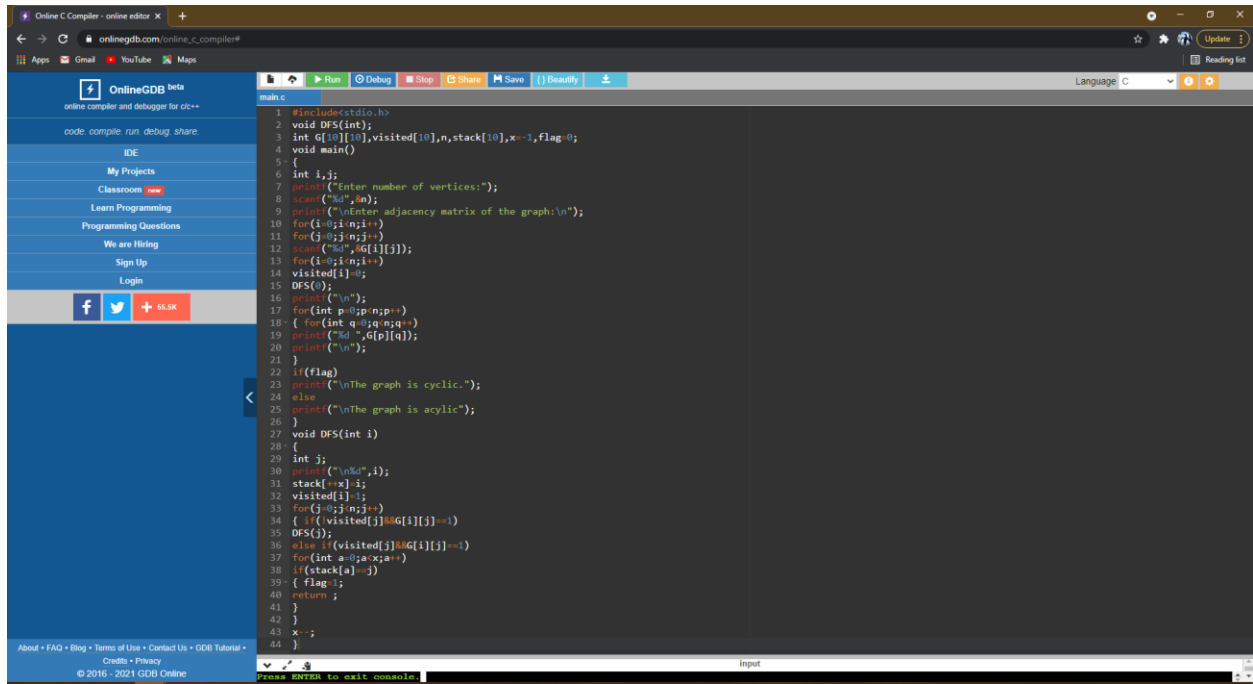
}

int main()
{ int a1[15],a2[15],n,i;

```

```
printf("Enter no of elements:");  
scanf("%d",&n);  
printf("Enter two arrays:(Both should contain %d elements):\n",n);  
printf("Enter Array 1:\n");  
for(i=0;i<n;i++)  
scanf("%d",&a1[i]);  
printf("Enter Array 2:\n");  
for(i=0;i<n;i++)  
scanf("%d",&a2[i]);  
printf("Median is %d", arrayMedian(a1, a2, n));  
getchar();  
return 0;  
}
```

OUTPUT FOR DFS CYCLIC

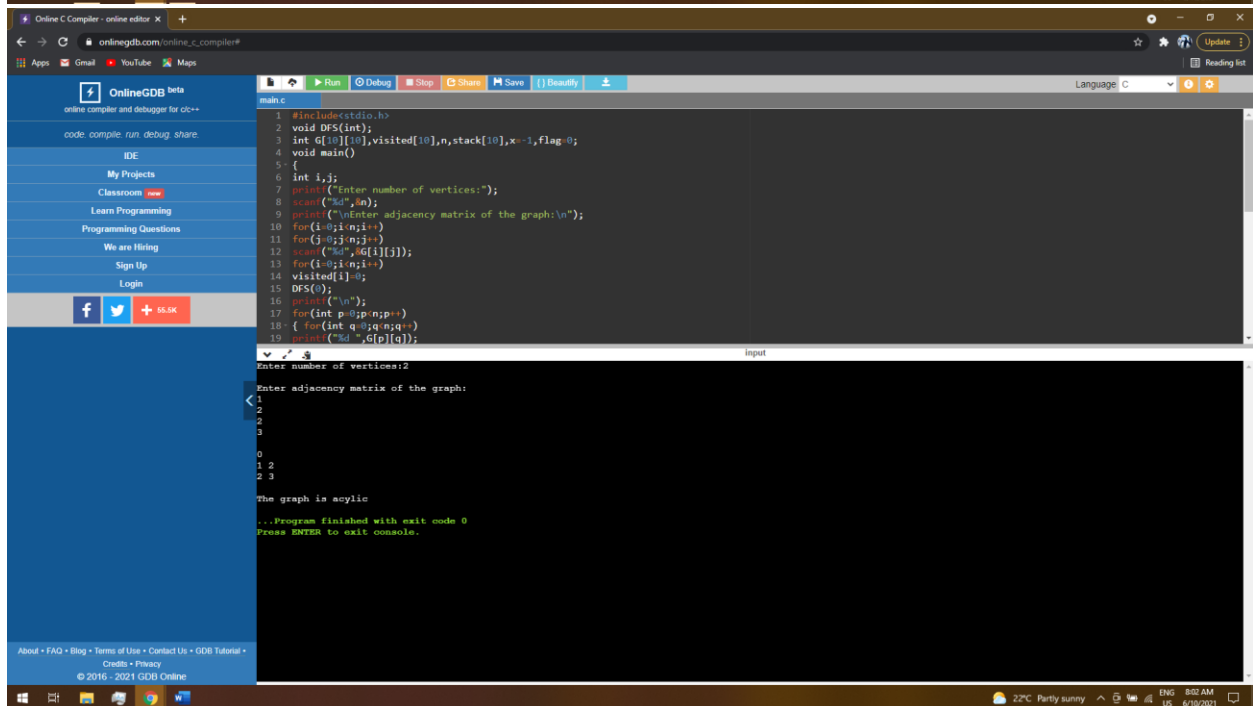


The screenshot shows the OnlineGDB interface with the following code in main.c:

```
1 #include<stdio.h>
2 void DFS(int);
3 int G[10][10],visited[10],n,stack[10],x=-1,flag=0;
4 void main()
5 {
6     int i,j;
7     printf("Enter number of vertices:");
8     scanf("%d",&n);
9     printf("\nEnter adjacency matrix of the graph:\n");
10    for(i=0;i<n;i++)
11        for(j=0;j<n;j++)
12            scanf("%d",&G[i][j]);
13    for(i=0;i<n;i++)
14        visited[i]=0;
15    DFS(0);
16    printf("\n");
17    for(int p=0;p<n;p++)
18        { for(int q=0;q<n;q++)
19            printf("%d ",G[p][q]);
20            printf("\n");
21        }
22    if(flag)
23        printf("\nThe graph is cyclic.");
24    else
25        printf("\nThe graph is acyclic.");
26 }
27 void DFS(int i)
28 {
29     int j;
30     printf("\n%d",i);
31     stack[x]=i;
32     visited[i]=1;
33     for(j=0;j<n;j++)
34         if((visited[j]==0&&G[i][j]==1)
35             DFS(j);
36     else if (visited[j]==1&&G[i][j]==1)
37         for(int a=0;a<x;a++)
38             if(stack[a]==j)
39                 { flag=1;
40                   return ;
41                 }
42 }
43 x--;
44 }
```

The output console shows the following text:

```
Press ENTER to exit console.
```



The screenshot shows the OnlineGDB interface with the same code as the previous screenshot. The output console shows the following text:

```
Enter number of vertices:2
Enter adjacency matrix of the graph:
1 0
2 0
3 0
0 1
1 2
2 3
The graph is acyclic
...Program finished with exit code 0
Press ENTER to exit console.
```

CODE FOR DFS CYCLIC

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

```
void DFS(int);
```

```
int G[10][10],visited[10],n,stack[10],x=-1,flag=0;
```

```

void main()
{
    int i,j;
    printf("Enter number of vertices:");
    scanf("%d",&n);
    printf("\nEnter adjacency matrix of the graph:\n");
    for(i=0;i<n;i++)
    for(j=0;j<n;j++)
    scanf("%d",&G[i][j]);
    for(i=0;i<n;i++)
    visited[i]=0;
    DFS(0);
    printf("\n");
    for(int p=0;p<n;p++)
    { for(int q=0;q<n;q++)
    printf("%d ",G[p][q]);
    printf("\n");
    }
    if(flag)
    printf("\nThe graph is cyclic.");
    else
    printf("\nThe graph is acyclic");
    }
    void DFS(int i)
    {
    int j;
    printf("\n%d",i);
    stack[++x]=i;
    visited[i]=1;

```

```
for(j=0;j<n;j++)
{ if(!visited[j]&&G[i][j]==1)
DFS(j);
else if(visited[j]&&G[i][j]==1)
for(int a=0;a<x;a++)
if(stack[a]==j)
{ flag=1;
return ;
}
}
x--;
}
```