

OUTPUT FOR MEDIAN

The screenshot shows a web-based IDE interface for OnlineGDB beta. The code in the editor is:

```
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 = m2;
11      m2 = a2[0];
12      break;
13    }
14    else if (j == n)
15    { m1 = m2;
16      m2 = a1[0];
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[5], a2[5], n, i;
34  printf("Enter no of elements:");
35 }
```

The terminal window shows the following interaction:

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

The status bar at the bottom right indicates: 23°C Party sunny 7:58 AM US 6/10/2021.

The screenshot shows two instances of the OnlineCCompiler interface. Both instances have the same code displayed in the editor:

```

main.c
13 }
14 else if (j == n)
15 { m1 = a2[j];
16 m2 = a1[0];
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[5],a2[5],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 getchar();
45 return 0;
46 }

```

The first instance shows the output window with the following interaction:

```

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

```

The second instance shows the output window with the following interaction:

```

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
...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 two instances of the OnlineGDB interface. The top instance displays the source code for a C program named 'main.c' that implements Depth-First Search (DFS) to detect cycles in a graph. The bottom instance shows the execution of this program in a terminal window.

Code (main.c):

```
#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("Enter 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)
                if(flag)
                    return;
    }
}
x--;
}
```

Execution Output:

```
Enter number of vertices:2
Enter adjacency matrix of the graph:
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--;
}
```