

main.c

```
4 int main() {
5     int arr[5]={17,2,39,45,5};
6     int m1,m2,temp;
7     m1=arr[0];
8     m2=arr[1];
9     int i;
10    if(m1<m2){
11        temp=m1;
12        m1=m2;
13        m2=temp;
14    }
15    for(i=1;i<6;i++){
16        if(arr[i]>m1){
17            m2=m1;
18            m1=arr[i];
19        }
20        else if(arr[i]>m2 && arr[i]<m1){
21            m2=arr[i];
22        }
23    }
24    printf("the sec max is %d",m2);
25    return 0;

```

Waiting for securepubads.g.doubleclick.net...

Output

```
/tmp/WzC5gNscQS.o
the sec max is 39
```

s Debug R-triangle.cpp

```
1 #include <stdio.h>
2 int main()
3 {
4     int i,j,rows;
5     printf("Input number of rows : ");
6     scanf("%d",&rows);
7     for(i=1;i<=rows;i++)
8     {
9         for(j=1;j<=i;j++)
10        {
11            printf("%d",j);
12        }
13    }
}
```

C:\Users\my pc\OneDrive\Desktop\data st\R-triangle.e... - X

```
Input number of rows : 6
1
12
123
1234
12345
123456

-----
Process exited after 3.112 seconds with return value 0
Press any key to continue . . .
```

Resources Compile Log Debug Find Results Close

Compilation results...

- Errors: 0

```
R-triangle.cpp hollow diamond.cpp [*] hollow 2.cpp
1 #include <stdio.h>
2 int main()
3 {
4     int n = 5, rows = 1, columns;
5     while (rows <= n) {
6         columns = n;
7         while (columns > rows) {
8             printf(" ");
9             columns--;
10    }
11    printf("*");
12    columns = 1;
13    while (columns < (rows - 1) * 2) {
14        printf(" ");
15        columns++;
16    }
17    if (rows == 1) {
18        printf("\n");
19    }
}
```

The screenshot shows a software interface with a code editor and a compilation results window.

Code Editor:

```
19 |     }
20 |     else {
21 |         printf("*\n");
22 |     }
23 |     rows++;
24 |
25 |     rows = n - 1;
26 |     while (rows >= 1) {
27 |         columns = n;
28 |         while (columns > rows) {
29 |             printf(" ");
30 |             columns--;
31 |         }
32 |         printf("**");
33 |         columns = 1;
34 |         while (columns < (rows - 1) * 2) {
35 |             printf(" ");
36 |             columns++;
37 |         }

```

Compiler Results:

```
Compilation results...
-----
- Errors: 0
- Warnings: 0
```

Screenshot of a C++ IDE showing code execution results.

The code in the editor is:

```
30     columns--;
31 }
32 printf("*");
33 columns = 1;
34 while (columns < (rows - 1) * 2) {
35     printf(" ");
36     columns++;
37 }
38 if (rows == 1) {
39     printf("\n");
40 }
41 else {
42     printf("*\n");
43 }
44 rows--;
45
46 return 0;
47 }
48 }
```

The output window shows the generated hollow diamond pattern:

```
*  
* *  
* * *  
* * * *  
* * * * *  
* * * * * *  
* * * * * * *  
-----  
Process exited after 0.05478 seconds with return value 0  
Press any key to continue . . .
```

A screenshot of a C++ development environment. The code editor displays a function named `spiralOfMatrix` which prints a spiral matrix pattern. The code uses nested loops and conditional statements to traverse the matrix in a spiral order. The IDE interface includes tabs for other files like `R-triangle.cpp`, `hollow diamond.cpp`, etc., and toolbars for compilation and resources.

```
1 #include <stdio.h>
2 #define R 4
3 #define C 5
4 void spiralOfMatrix(int enrow, int encol, int arr1[R][C])
5 {
6     int i, rowind = 0, colind = 0;
7     while (rowind < enrow && colind < encol)
8     {
9         for (i = colind; i < encol; ++i)
10        {
11            printf("%d ", arr1[rowind][i]);
12        }
13        rowind++;
14        for (i = rowind; i < enrow; ++i)
15        {
16            printf("%d ", arr1[i][encol-1]);
17        }
18        encol--;
19        if (rowind < enrow)
```

The screenshot shows a C++ development environment with the following details:

- Toolbar:** Standard file operations (File, Edit, Search, View, Project, Execute, Tools, AStyle, Window, Help), build controls (Compile, Run, Stop, Build, Clean, Rebuild), and navigation (Search, Find, Replace, Go To, Jump, Symbols).
- Status Bar:** TDM-GCC 4.9.2 64-bit Release.
- Project Tab:** Shows multiple projects: R-triangle.cpp, hollow diamond.cpp, [*] hollow 2.cpp, zig zag.cpp, and spiral matrix.cpp. The hollow diamond.cpp tab is active.
- Code Editor:** Displays the following C++ code for printing a hollow diamond pattern:

```
19 |     if ( rowind < enrow)
20 |     {
21 |         for (i = encol-1; i >= colind; --i)
22 |         {
23 |             printf("%d ", arr1[enrow-1][i]);
24 |         }
25 |         enrow--;
26 |
27 |         if (colind < encol)
28 |         {
29 |             for (i = enrow-1; i >= rowind; --i)    [
30 |                 printf("%d ", arr1[i][colind]);
31 |             }
32 |             colind++;
33 |
34 |         }
35 |
36 |     }
37 | int main()
```

The code uses nested loops to print a hollow diamond shape. It prints the outer boundary of the diamond and leaves the inner part empty. The arr1 array is used to store the values of the diamond.

Tool Buttons: Compiler, Resources, Compile Log, Debug, Find Results, Close.

Bottom Status: Compilation results... About Compilation

IDM-GCC 4.9.2 64-bit Release

```
Project Classes Debug R-triangle.cpp hollow diamond.cpp [*] hollow 2.cpp zig zag.cpp spiral matrix.cpp
37 int main()
38 {
39     int i,j;
40     int arr1[R][C] = { {1, 2, 3, 4, 5},
41                         {6, 7, 8, 9, 10},
42                         {11, 12, 13, 14, 15},
43                         {16, 17, 18, 19, 20}
44                     };
45     printf("The given array in matrix form is : \n");
46     for(i = 0; i < R; i++)
47     {
48         for (j=0;j<C;j++)
49         {
50             printf("%d ", arr1[i][j]);
51         }
52         printf("\n");
53     }
54     printf("The spiral form of above matrix is: \n");
55     spiralOfMatrix(R, C, arr1);
```

Compiler Resources Compile Log Debug Find Results Close

Abort Compilation Compilation results...

The screenshot shows a C++ development environment with the following details:

- Toolbar:** Standard icons for file operations, search, and navigation.
- Menu Bar:** File, Edit, View, Analyze, Window, Help.
- Tool Bar:** Includes icons for project management, build, and run.
- Project Explorer:** Shows files: R-triangle.cpp, hollow diamond.cpp, hollow 2.cpp, zig zag.cpp, and spiral matrix.cpp.
- Code Editor:** Displays the following C++ code:

```
39 | int i,j;
40 |     int arr1[R][C] = { {1, 2, 3, 4, 5},
41 |         {6, 7, 8, 9, 10},
42 |         {11, 12, 13, 14, 15},
43 |         {16, 17, 18, 19, 20}
44 |     };
45 |     printf("The given array in matrix form is : \n");
46 |     for(i = 0; i < R; i++)
47 |     {
48 |         for (j=0;j<C;j++)
49 |         {
50 |             printf("%d ", arr1[i][j]);
51 |         }
52 |         printf("\n");
53 |     }
54 |     printf("The spiral form of above matrix is: \n");
55 |     spiralOfMatrix(R, C, arr1);
56 |     return 0;
57 | }
```

- Output Window:** Shows the terminal output of the program execution.
- Compilation Results:** Shows compilation results with 0 errors and 0 warnings.

```
The given array in matrix form is :
1 2 3 4 5
6 7 8 9 10
11 12 13 14 15
16 17 18 19 20
The spiral form of above matrix is:
1 2 3 4 5 10 15 20 19 18 17 16 11 6 7 8 9 14 13 12
-----
Process exited after 0.06813 seconds with return value 0
Press any key to continue . . .
```

Compilation results...

- Errors: 0
- Warnings: 0

A screenshot of a Windows desktop environment. In the foreground, there is a code editor window titled "main.c". The code editor interface includes a toolbar with icons for file operations, a status bar at the bottom, and a search bar. The main area displays C code for printing a diamond pattern. The code uses nested loops to calculate the number of spaces and asterisks for each row based on the input number. The output pane shows the result of running the program with the input value 6, displaying a diamond shape made of asterisks.

```
1 #include<stdio.h>
2 int main() {
3     int n, i, k;
4     printf("Enter Number: ");
5     scanf("%d", &n);
6     for (int i = 1; i <= n; i = i + 2)
7     {
8         for (k = 1; k <= (n - i) / 2; ++k)
9         {
10             printf(" ");
11         }
12         for (k = 1; k <= i; ++k)
13         {
14             printf("*");
15         }
16         printf("\n");
17     }
18     for (i = n / 2 + n / 2 - 1; i >= 1; i = i - 2)
19     {
20         for (k = 1; k <= (n - i) / 2; ++k)
21         {
22             printf(" ");
23         }
24     }
}
```

Output

```
/tmp/rLMmxusX87.o
Enter Number: 6
*
 ***
 *****
 *****
 ***
 *
|
```

The screenshot shows a code editor interface with a tab bar at the top containing icons for various languages: Python, C/C++, C, C++, C, C, C, JS, Go, and PHP. The active tab is labeled "main.c". The main workspace displays the following C code:

```
main.c
13     {
14         printf("*");
15     }
16     printf("\n");
17 }
18 for (i = n / 2 + n / 2 - 1; i >= 1; i = i - 2)
19 {
20     for (k = 1; k <= (n - i) / 2; ++k)
21     {
22         printf(" ");
23     }
24     for (k = 1; k <= i; ++k)
25     {
26         printf("*");
27     }
28     printf("\n");
29 }
30
31 return 0;
32
33 }
34
```

The "Run" button is highlighted in blue. To the right of the code editor is an "Output" panel showing the program's execution results:

```
^ /tmp/rLMMxusX87.o
Enter Number: 6
*
***  
*****  
*****  
***  
*
```

At the bottom right of the screen, there are additional UI elements: "Activate Window", "Go to Settings to ad", and a small icon area.