

## Pascal's Triangle Coding Solution

**HATFD1035**

### **Problem Statement:**

Print Pascal's Triangle Write a program to generate and print the first n rows of Pascal's triangle without using built-in math or array functions. For n = 5, the output should be:

```
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
```

### **SOLUTION:**

```
echo -n "Enter n: "
read n

for ((row=0; row<n; row++)); do
    # print leading spaces for alignment
    for ((space=0; space<n-row; space++)); do
        echo -n " "
    done

    # calculate values for the row
    value=1
    for ((col=0; col<=row; col++)); do
        # Print the current value
        echo -n "$value "
```

```
# calculate the next value in the row
value=$((value * (row - col) / (col + 1)))

done

echo

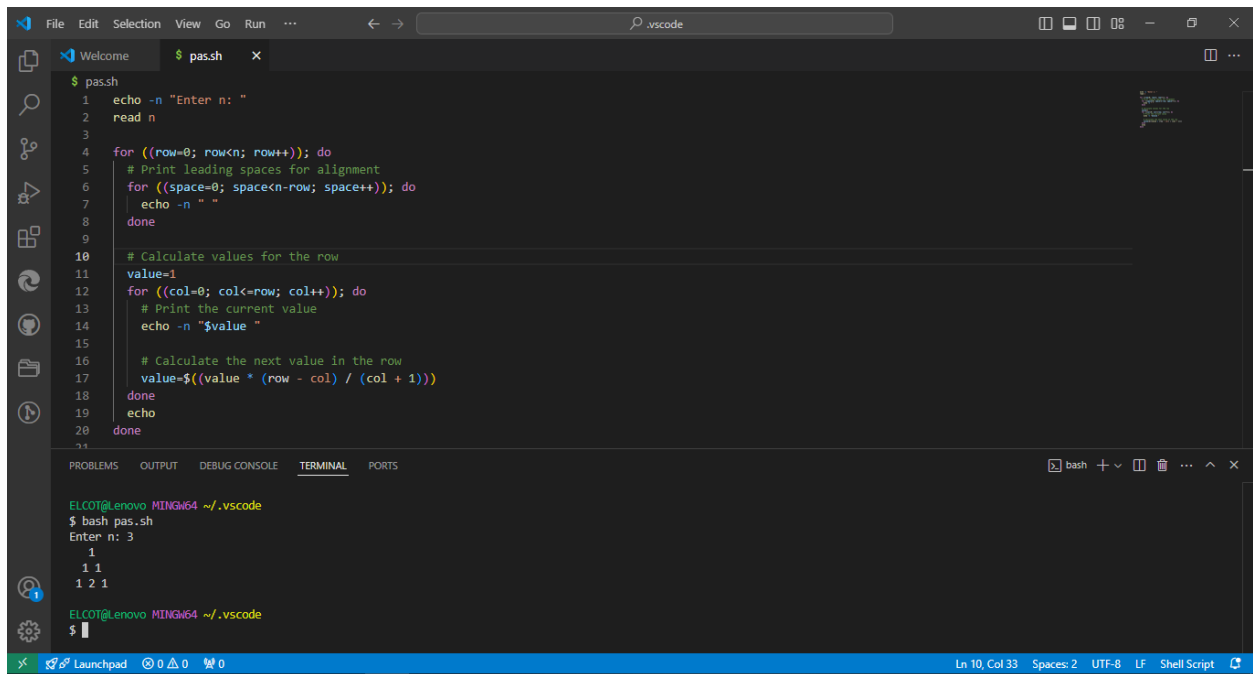
done
```

### Sample Inputs/Outputs

Sample Input 1: Enter n: 3

Output:

```
1
1 1
1 2 1
```



The screenshot shows a VS Code editor window with a file named `pas.sh`. The script contains the following code:

```
1 echo -n "Enter n: "
2 read n
3
4 for ((row=0; row<n; row++)); do
5     # Print leading spaces for alignment
6     for ((space=0; space<n-row; space++)); do
7         echo -n " "
8     done
9
10    # Calculate values for the row
11    value=1
12    for ((col=0; col<=row; col++)); do
13        # Print the current value
14        echo -n "$value "
15
16        # Calculate the next value in the row
17        value=$((value * (row - col) / (col + 1)))
18    done
19    echo
20 done
```

Below the editor, the `TERMINAL` panel shows the execution of the script:

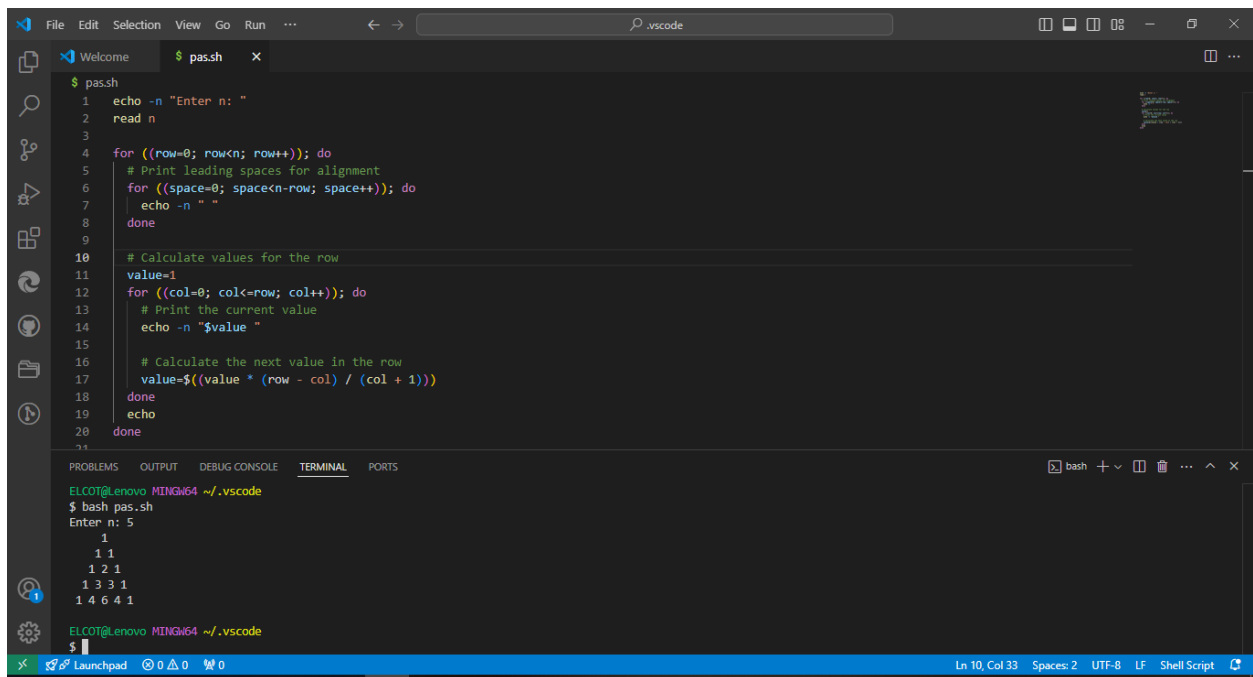
```
ELCOT@Lenovo MINGW64 ~/vscode
$ bash pas.sh
Enter n: 3
1
1 1
1 2 1
$
```

The status bar at the bottom indicates the file is at line 10, column 33, with 2 spaces, UTF-8 encoding, LF line endings, and is a shell script.

Sample Input 2: Enter n: 5

Output:

```
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
```



The screenshot shows a Visual Studio Code editor window with a dark theme. The editor has a sidebar on the left with icons for Explorer, Search, Source Control, Run and Debug, and Extensions. The main editor area displays a shell script named `passh`. The script uses nested loops to calculate and print Pascal's triangle for a given input `n`. The script includes comments for each step: reading input, calculating leading spaces, calculating values for each row, and printing the current value. Below the editor, a terminal window is open, showing the execution of the script. The terminal prompt is `ELCOT@lenovo MINGW64 ~/vscode`. The user enters `bash passh`, and the script prompts `Enter n: 5`. The output of the script is displayed in the terminal, matching the sample output provided above.

```
$ passh
1 echo -n "Enter n: "
2 read n
3
4 for ((row=0; row<n; row++)); do
5     # Print leading spaces for alignment
6     for ((space=0; space<n-row; space++)); do
7         echo -n " "
8     done
9
10    # Calculate values for the row
11    value=1
12    for ((col=0; col<=row; col++)); do
13        # Print the current value
14        echo -n "$value "
15
16        # Calculate the next value in the row
17        value=$((value * (row - col) / (col + 1)))
18    done
19    echo
20 done
```

```
ELCOT@lenovo MINGW64 ~/vscode
$ bash passh
Enter n: 5
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
$
```

1  
1 1  
1 2 1  
1 3 3 1  
1 4 6 4 1  
1 5 10 10 5 1

```

1  echo -n "Enter n: "
2  read n
3
4  for ((row=0; row<n; row++)); do
5      # Print leading spaces for alignment
6      for ((space=0; space<n-row; space++)); do
7          echo -n " "
8      done
9
10     # Calculate values for the row
11     value=1
12     for ((col=0; col<=row; col++)); do
13         # Print the current value
14         echo -n "$value "
15
16         # Calculate the next value in the row
17         value=$((value * (row - col) / (col + 1)))
18     done
19     echo
20 done

```

```

$ bash pas.sh
Enter n: 6
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1

```