

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
```

PROGRAM:

```
print_spaces() {
    for ((j=0; j<$1; j++)); do
        echo -n " "
    done
}

while true; do
    echo -n "Enter n (positive integer): "
    read rows
    if [[ "$rows" =~ ^[0-9]+$ ]] && [ "$rows" -gt 0 ]; then
        break
    else
        echo "Invalid input. Please enter a positive integer."
    fi
done

max_num=$(( (1 << rows) / 2 )) # Largest number in the last row
alignment_width=${#max_num}

for ((i=0; i<rows; i++)); do
    # Print leading spaces for alignment
    print_spaces $((rows - i))
    num=1
    for ((k=0; k<=i; k++)); do
```

```
printf "%${alignment_width}d " "$num"
```

```
num=$((num * (i - k) / (k + 1)))
```

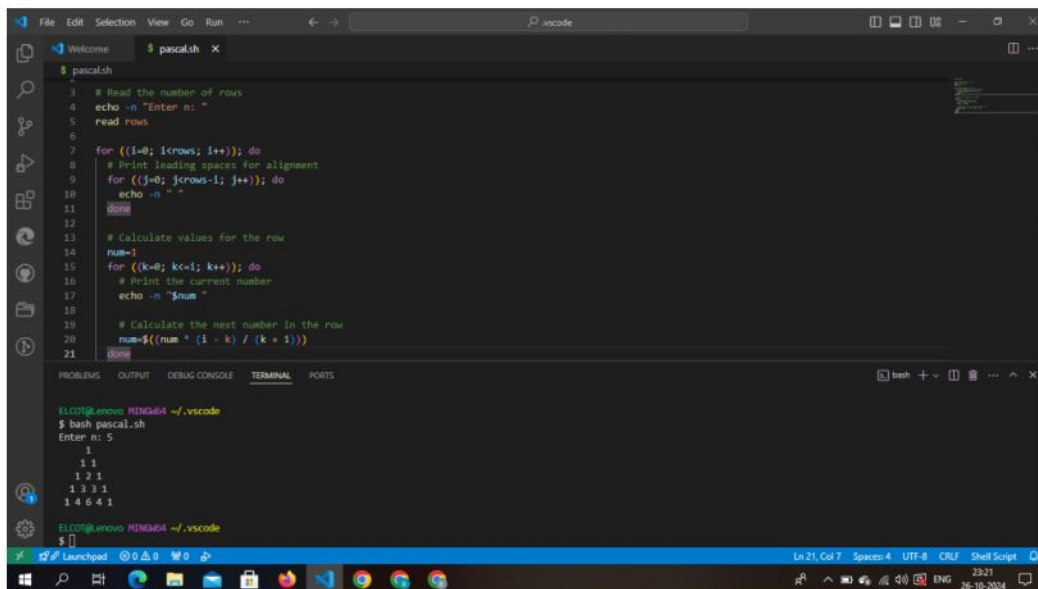
```
done
```

```
echo
```

```
done
```

SAMPLE INPUT /OUTPUT:

N=5



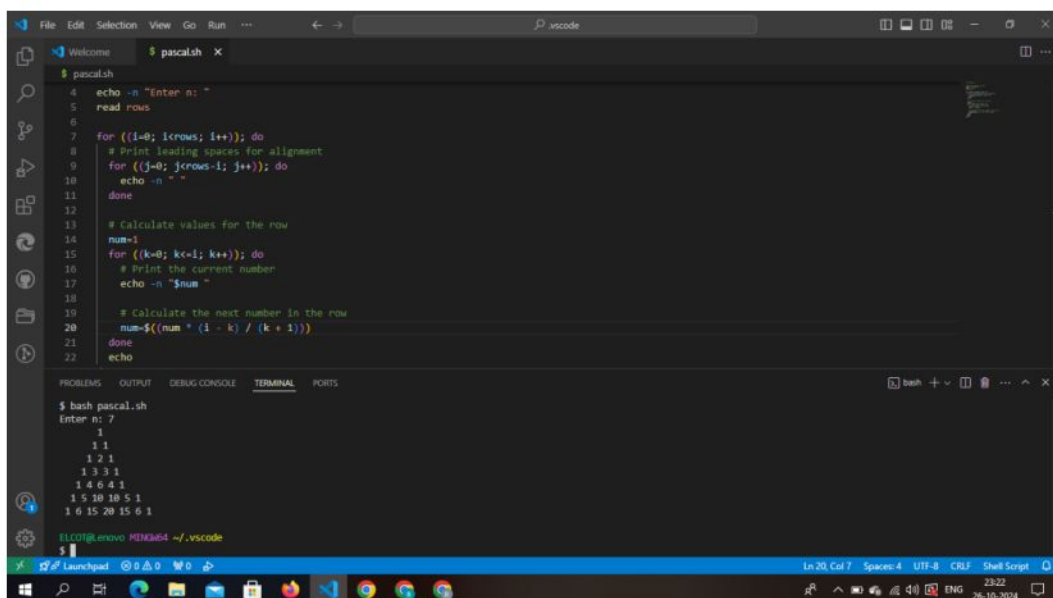
The screenshot shows a VS Code editor with a file named `pascal.sh`. The script contains the following code:

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

The terminal output shows the script being executed with `rows=5`, resulting in the following Pascal's triangle:

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

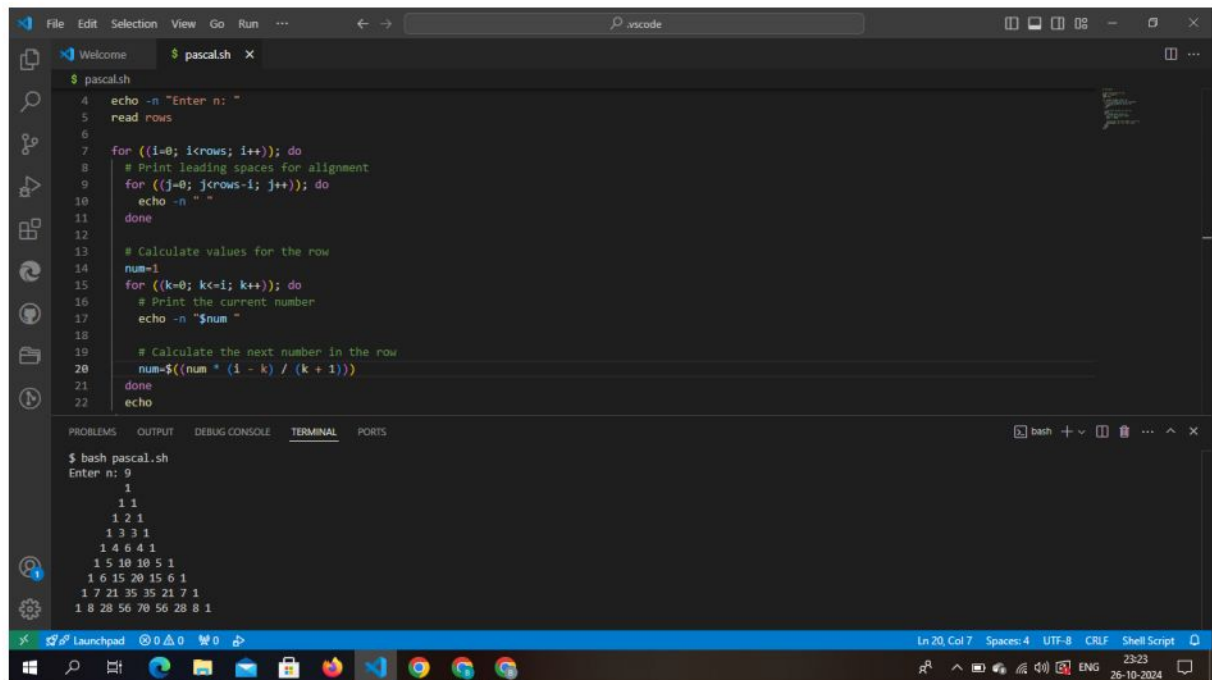
N=7



The screenshot shows the same VS Code editor with the `pascal.sh` script. The terminal output shows the script being executed with `rows=7`, resulting in the following Pascal's triangle:

```
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
1 6 15 20 15 6 1
```

N=9



The image shows a Visual Studio Code (VS Code) editor window with a dark theme. The editor is open to a file named `pascal.sh`. The script is a Bash script that generates Pascal's triangle for a given number of rows (`n`). The script uses nested loops to calculate the values for each row and column, and it prints the triangle with leading spaces for alignment. The script is as follows:

```
4 echo -n "Enter n: "
5 read rows
6
7 for ((i=0; i<rows; i++)); do
8     # Print leading spaces for alignment
9     for ((j=0; j<rows-i; j++)); do
10         echo -n " "
11     done
12
13     # Calculate values for the row
14     num=1
15     for ((k=0; k<=i; k++)); do
16         # Print the current number
17         echo -n "$num "
18
19         # Calculate the next number in the row
20         num=$((num * (i - k) / (k + 1)))
21     done
22     echo
```

The terminal window at the bottom shows the output of the script. It prompts the user to enter a value for `n`, and the user enters `9`. The output is a Pascal's triangle with 9 rows, where each row contains numbers from 1 to the row number, and the numbers are aligned to the right of the row.

```
$ bash pascal.sh
Enter n: 9
      1
     1 1
    1 2 1
   1 3 3 1
  1 4 6 4 1
 1 5 10 10 5 1
1 6 15 20 15 6 1
1 7 21 35 35 21 7 1
1 8 28 56 70 56 28 8 1
```

The status bar at the bottom of the VS Code window shows the current line and column (Ln 20, Col 7), the number of spaces (Spaces: 4), the encoding (UTF-8), the line ending (CRLF), and the file type (Shell Script).