## 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
```

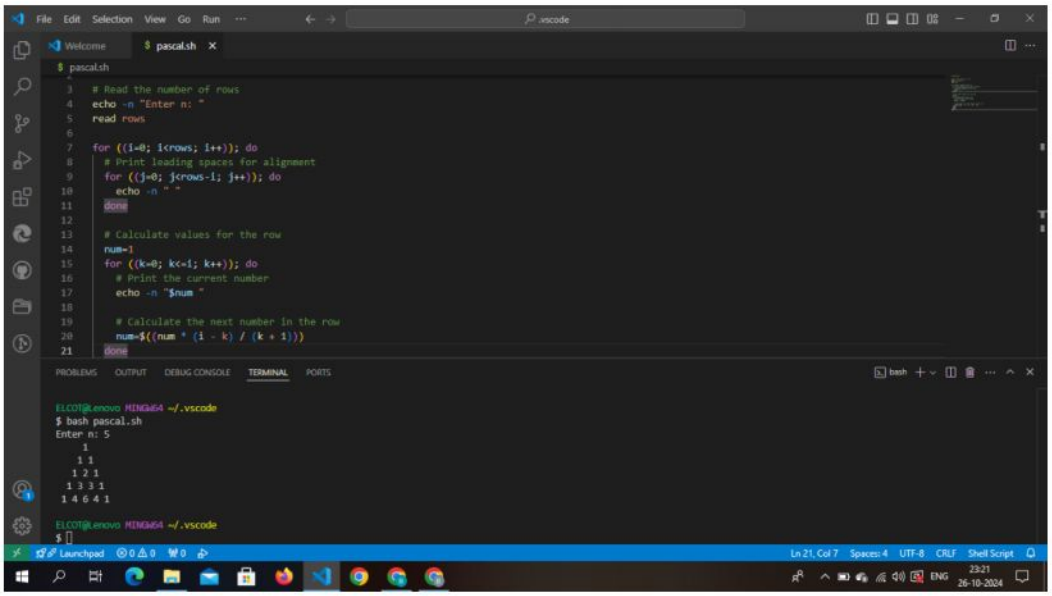
**PROGRAM:**

```
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
```

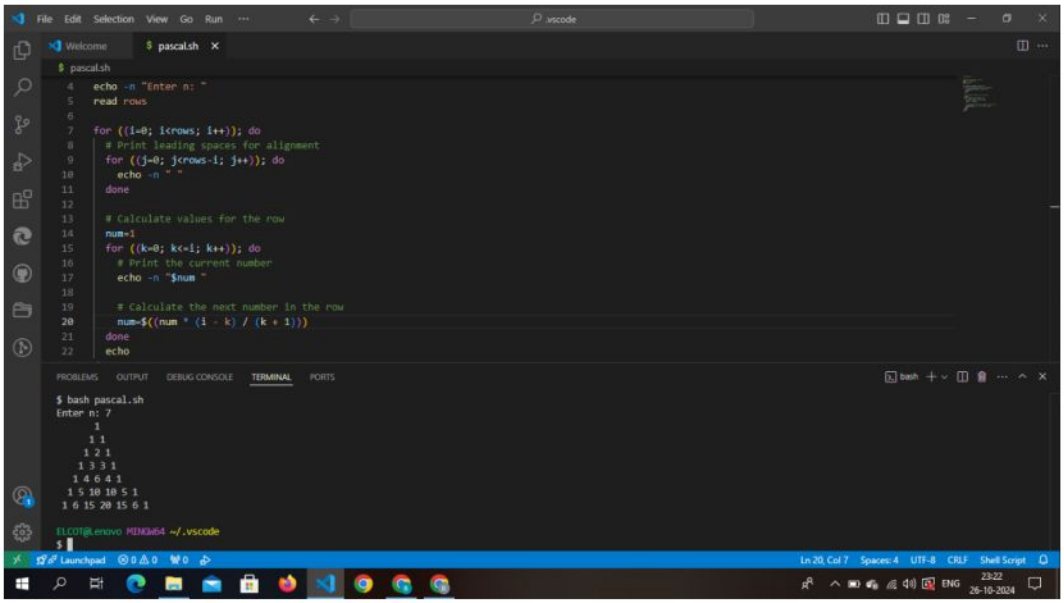
```
    printf "%${alignment_width}d " "$num"
    num=$((num * (i - k) / (k + 1)))
  done
  echo
done
```

**SAMPLE INPUT /OUTPUT:**

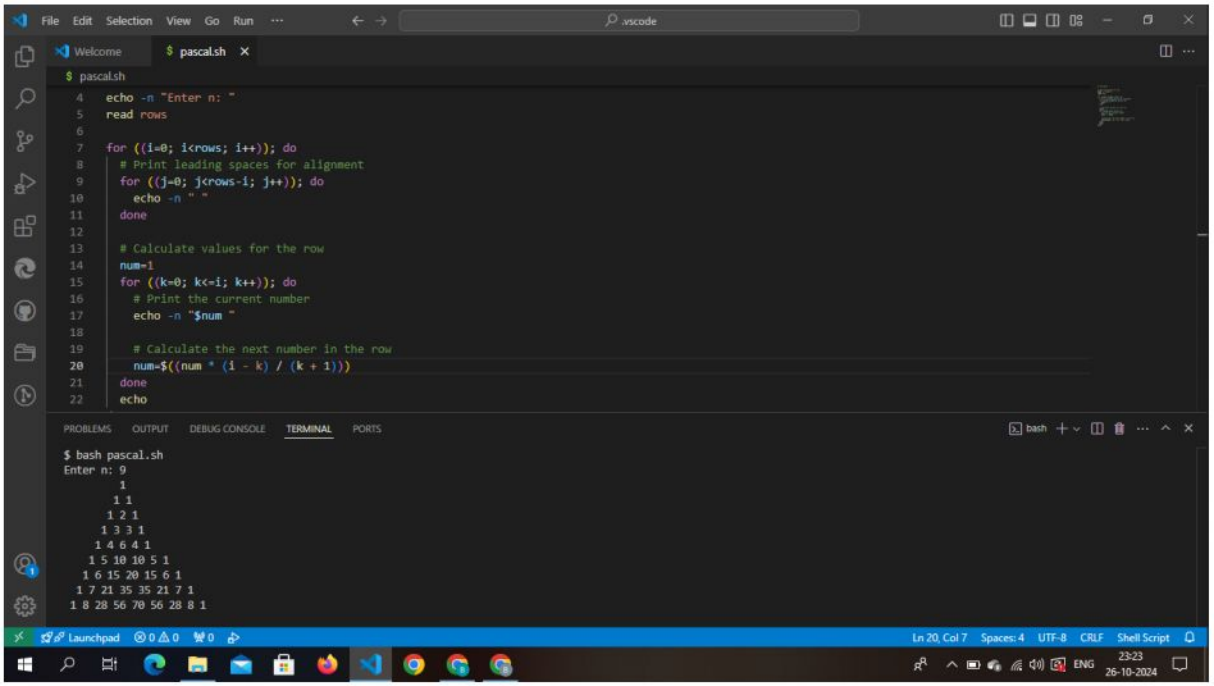
**N=5**




**N=7**



**N=9**


File Edit Selection View Go Run
.vscode
Welcome
pascal.sh
$ pascal.sh
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
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
bash
$ 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
Launchpad
Ln 20, Col 7 Spaces: 4 UTF-8 CRLF Shell Script
ENG
23:23
26-10-2024