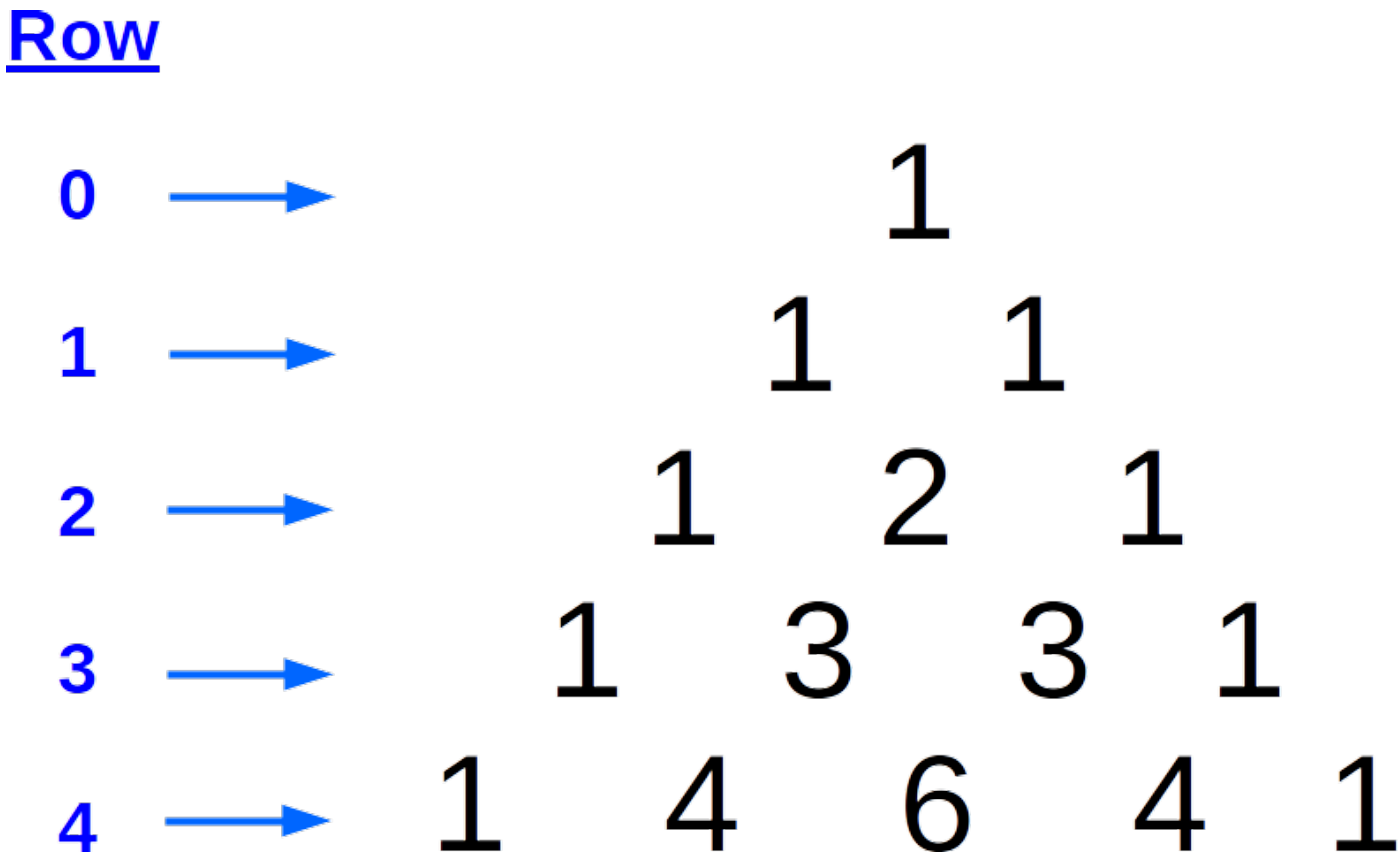
MP3

MP 3 - Pascal's triangle

Printing a row from Pascal's triangle

In this MP, you will implement a C program to print a row from Pascal's triangle. Pascal's triangle is an array that consists of binomial coefficients and the figure below shows the first five rows of Pascal's triangle.



A row in the Pascal' triangle contains all the **coefficients** of expanding the polynomial $(1+x)^n$, where n corresponds to the index of row in Pascal's triangle.

Details

We refer to the kth *binomial coefficient in the* nth row by $\binom{n}{k}$ where the coefficient can be computed by the following formula:

$$\binom{n}{k} = \frac{n!}{k!\,(n-k)!} = \frac{n\,(n-1)\,(n-2)\ldots(n-k+1)}{1\times 2\times 3\times\cdots\times k} = \prod_{i=1}^{k}\frac{n+1-i}{i}$$

where $\binom{n}{0} := 1 \quad \forall n \geq 0$. Note that both k, n start from 0 and $0 \leq k \leq n$.

The program first asks the user to enter a row index (this part has been provided to you) and your task is to calculate all coefficients in that row and use the standard function `printf` to print out the coefficients. Refer to **http://www.cplusplus.com/reference/cstdio/printf/ (http://www.cplusplus.com/reference/cstdio/printf/)** for using printf.

The output should have a space between successive numbers, for example

```
Enter the row index: 3
1 3 3 1
```

As the coefficients are integral, you should **NOT** use float or double to store the coefficients. Instead, **to prevent the number overflow** during computing:

- you **should** use **unsigned long** for the coefficients.
- you **should** use the equation with $\prod$

You can assume that the input row index will be $\leq$ 40.

Building and Testing

To compile your program, type the following command:

```
gcc -Wall -g mp3.c -std=c99 -o mp3
```

The `-Wall` turns on compiler's warning messages and you have to fix all warnings and errors in compilation.

If the compilation succeeds, you will get a binary called **mp3**. To execute your program, type :

```
./mp3
```

Here are some sample outputs for verifying your program:

You must follow this format exactly as we will use autograder. You can have space or new line at the end or not.

```
Enter the row index: 0
1
```

```
Enter the row index: 1
1 1
```

```
Enter the row index: 2
1 2 1
```

```
Enter the row index: 3
1 3 3 1
```

```
Enter the row index: 4
1 4 6 4 1
```

Grading Rubric

Functionality (90%)

The program outputs the correct results for various inputs.

Style, comments, clarity, and write-up (10%)

- Comments and Style - 5%
  - If your code has the correct output but uses double or float to store values, style points will be deducted.
  - Functionality points may be lost if using double/float results in incorrect output.
- Intro Paragraph - 5%

If your code does not compile you will get 0