Autumn 2022, ISTM, FCU-Purdue 2+2 ECE Program
ISTM116 Programming Applications for Engineers, Quiz 2

Use file name "quiz2_Dxxxxxxx_1.c" for Question 1 and file name "quiz2_Dxxxxxxx_2.c" for Question 2 for your source, where "Dxxxxxxx" is your student ID. When you finish a question, upload the source code file to the instructor's computer.

1. (50 points) You may start with program skeleton **quiz2_skeleton_1.c** and change the file name to **quiz2_Dxxxxxxx_1.c**. Consider the expansion of $(a+b)^n$. If the coefficients of $(a+b)^k$ for $0 \le k \le n$, are listed in a row, we obtain a triangle, called **Pascal triangle**, as below for n=5:

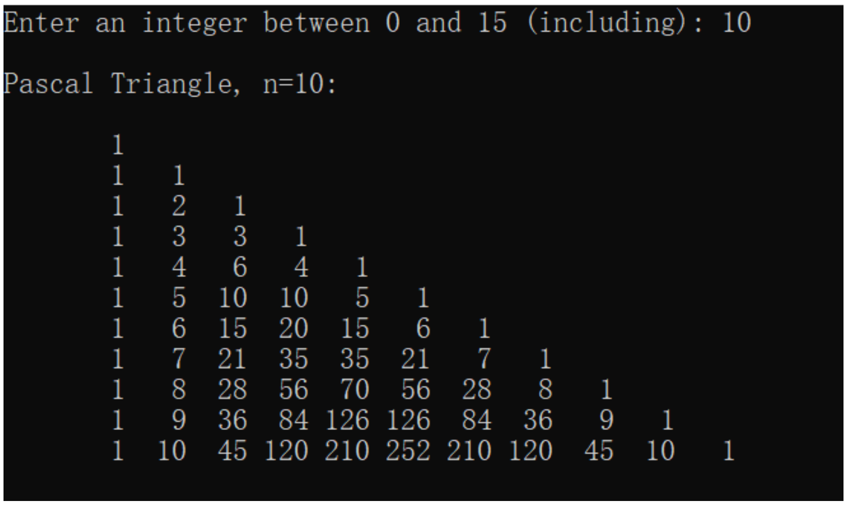
k=0	1
k=1	1 1
k=2	1 2 1
k=3	1 3 3 1
k=4	1 4 6 4 1
k=5	1 5 10 10 5 1

From the mathematical point of view, the coefficients of expanding of $(a+b)^n$ is the following sequence:

$$\binom{n}{0}, \binom{n}{1}, \binom{n}{2}, \cdots, \binom{n}{n-2}, \binom{n}{n-1}, \binom{n}{n},$$

where $\binom{n}{k}$ is the combination function, i.e., $\binom{n}{k} = \frac{n!}{k!(n-k)!}$. Write a C program to input an integer n between 0 and 10 (including) and print the Pascal triangle of coefficients of $(a+b)^n$. Align the Pascal triangle to the left-hand-side. You are required to define a *recursive function* **int** factorial(**int** k) to compute k! and function **int** C(**int** k, **int** j) to compute combination function $\binom{k}{j}$.

Example of program execution:



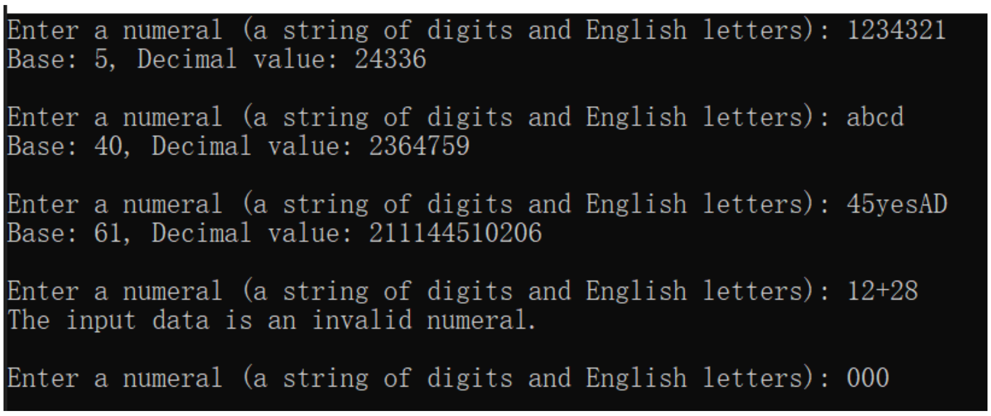
```
Enter an integer between 0 and 15 (including): 10

Pascal Triangle, n=10:

       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
       1   9  36  84 126 126  84  36   9   1
       1  10  45 120 210 252 210 120  45  10   1
```

(continue to the next page).

2. (50 points) You may start with program skeleton **quiz2_skeleton_2.c** and change the file name to **quiz2_Dxxxxxxx_2.c**. The following table is the digit-value mapping for base-62 numerals. Write a C program to repeatedly enter a string of digits and English letters and convert it to a decimal value with the ***smallest possible base***. For example, "1234321" is converted to a decimal value as a base-5 numeral; "abcd" is converted to a decimal value as a base-40 numeral; "45yesAD" is converted to a decimal value as a base-61 numeral. For each iteration, output the smallest possible base and the converted decimal value. The program terminates when the input is a string of 0's.

digit	value	digit	value	digit	value	digit	value	digit	value
0	0	E	14	S	28	g	42	u	56
1	1	F	15	T	29	h	43	v	57
2	2	G	16	U	30	i	44	w	58
3	3	H	17	V	31	j	45	x	59
4	4	I	18	W	32	k	46	y	60
5	5	J	19	X	33	l	47	z	61
6	6	K	20	Y	34	m	48		
7	7	L	21	Z	35	n	49		
8	8	M	22	a	36	o	50		
9	9	N	23	b	37	p	51		
A	10	O	24	c	38	q	52		
B	11	P	25	d	39	r	53		
C	12	Q	26	e	40	s	54		
D	13	R	27	f	41	t	55		

Report an error message, if the numeral contains a non-alphanumerical character. Note that function strlen(str) in "<string.h>" returns the length of string str. Example of program execution:



```
Enter a numeral (a string of digits and English letters): 1234321
Base: 5, Decimal value: 24336

Enter a numeral (a string of digits and English letters): abcd
Base: 40, Decimal value: 2364759

Enter a numeral (a string of digits and English letters): 45yesAD
Base: 61, Decimal value: 211144510206

Enter a numeral (a string of digits and English letters): 12+28
The input data is an invalid numeral.

Enter a numeral (a string of digits and English letters): 000
```