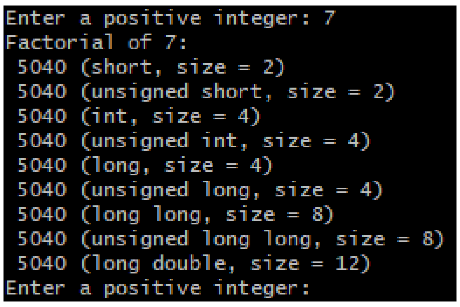
(60-140) Lab Exercises #7

— Working with basic types

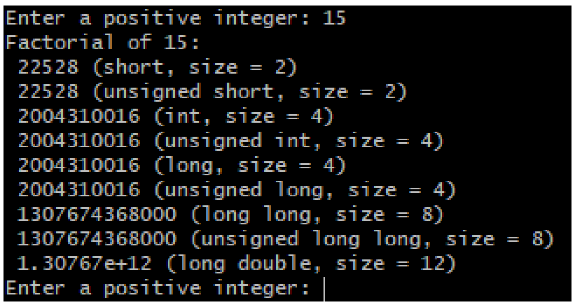
November 6, 2017

1. Enhancement to Lab #5

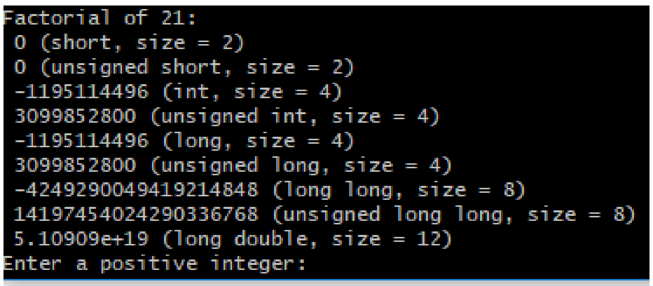
   (a) When running "l5_factorial.c", the program prints only one result of factorial calculation. This exercise requires an enhancement to "l5_factorial.c". The new version needs to list the results of factorial calculation for all the integer types of C99, from `short int` to `unsigned long long int`, together with information of the corresponding data type and size. The following output samples illustrate some executions of the enhancement program:



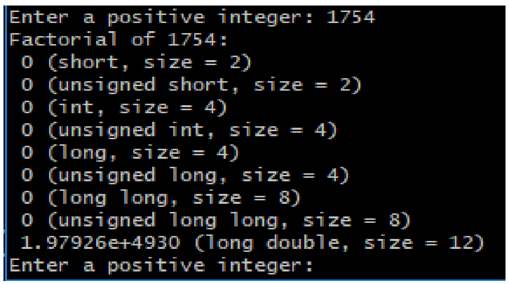
```
Enter a positive integer: 7
Factorial of 7:
 5040 (short, size = 2)
 5040 (unsigned short, size = 2)
 5040 (int, size = 4)
 5040 (unsigned int, size = 4)
 5040 (long, size = 4)
 5040 (unsigned long, size = 4)
 5040 (long long, size = 8)
 5040 (unsigned long long, size = 8)
 5040 (long double, size = 12)
Enter a positive integer:
```



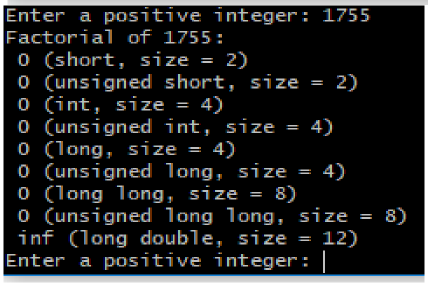
```
Enter a positive integer: 15
Factorial of 15:
 22528 (short, size = 2)
 22528 (unsigned short, size = 2)
 2004310016 (int, size = 4)
 2004310016 (unsigned int, size = 4)
 2004310016 (long, size = 4)
 2004310016 (unsigned long, size = 4)
 1307674368000 (long long, size = 8)
 1307674368000 (unsigned long long, size = 8)
 1.30767e+12 (long double, size = 12)
Enter a positive integer: |
```



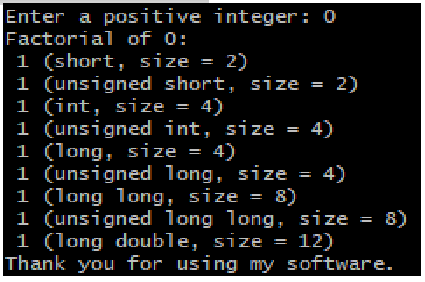
```
Factorial of 21:
 0 (short, size = 2)
 0 (unsigned short, size = 2)
 -1195114496 (int, size = 4)
 3099852800 (unsigned int, size = 4)
 -1195114496 (long, size = 4)
 3099852800 (unsigned long, size = 4)
 -4249290049419214848 (long long, size = 8)
 14197454024290336768 (unsigned long long, size = 8)
 5.10909e+19 (long double, size = 12)
Enter a positive integer:
```



```
Enter a positive integer: 1754
Factorial of 1754:
 0 (short, size = 2)
 0 (unsigned short, size = 2)
 0 (int, size = 4)
 0 (unsigned int, size = 4)
 0 (long, size = 4)
 0 (unsigned long, size = 4)
 0 (long long, size = 8)
 0 (unsigned long long, size = 8)
 1.97926e+4930 (long double, size = 12)
Enter a positive integer:
```



```
Enter a positive integer: 1755
Factorial of 1755:
 0 (short, size = 2)
 0 (unsigned short, size = 2)
 0 (int, size = 4)
 0 (unsigned int, size = 4)
 0 (long, size = 4)
 0 (unsigned long, size = 4)
 0 (long long, size = 8)
 0 (unsigned long long, size = 8)
 inf (long double, size = 12)
Enter a positive integer: |
```



```
Enter a positive integer: 0
Factorial of 0:
 1 (short, size = 2)
 1 (unsigned short, size = 2)
 1 (int, size = 4)
 1 (unsigned int, size = 4)
 1 (long, size = 4)
 1 (unsigned long, size = 4)
 1 (long long, size = 8)
 1 (unsigned long long, size = 8)
 1 (long double, size = 12)
Thank you for using my software.
```

   (b) Identify when each of the integer types starts to produce incorrect results by trying out your program with those numbers, and submit online your explanations for the failure of each of the integer types.

2!	= 2	13!	= 6,227,020,800	100!	= $9.332621544\times10^{157}$
3!	= 6	14!	= 87,178,291,200	450!	= $1.733368733\times10^{1000}$
4!	= 24	15!	= 1,307,674,368,000	1,000!	= $4.023872601\times10^{2567}$
5!	= 120	16!	= 20,922,789,888,000	3,249!	= $6.412337688\times10^{10000}$
6!	= 720	17!	= 355,687,428,096,000	10,000!	= $2.846259681\times10^{35659}$
7!	= 5,040	18!	= 6,402,373,705,728,000	25,206!	= $1.205703438\times10^{100000}$
8!	= 40,320	19!	= 121,645,100,408,832,000	100,000!	= $2.824229408\times10^{456573}$
9!	= 362,880	20!	= 2,432,902,008,176,640,000	205,023!	= $2.503898932\times10^{1000004}$
10!	= 3,628,800	25!	= 1.551121004×10^{25}	1,000,000!	= $8.263931688\times10^{5565708}$
11!	= 39,916,800	50!	= 3.041409320×10^{64}	10^{100}!	= $10^{9.956570552\times10^{101}}$
12!	= 479,001,600	70!	= $1.197857167\times10^{100}$		

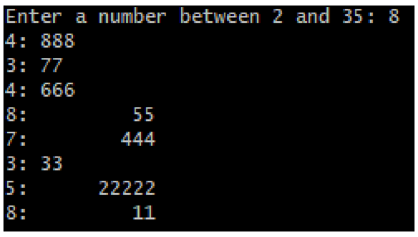
   (c) Which of the other data types can you use to find out the exact factorial number that causes `unsigned long long int` to fail? What is this number?

   (d) Save your enhanced program to "l7 factorial.c", and submit it online.

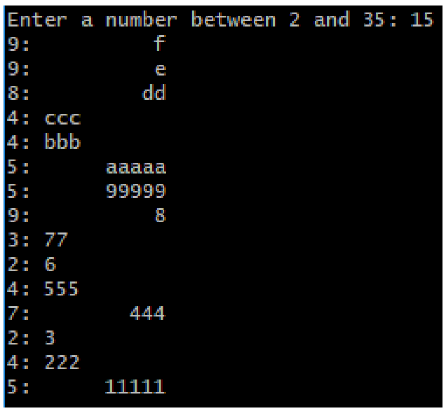
-

2. Enhancement to Lab #6

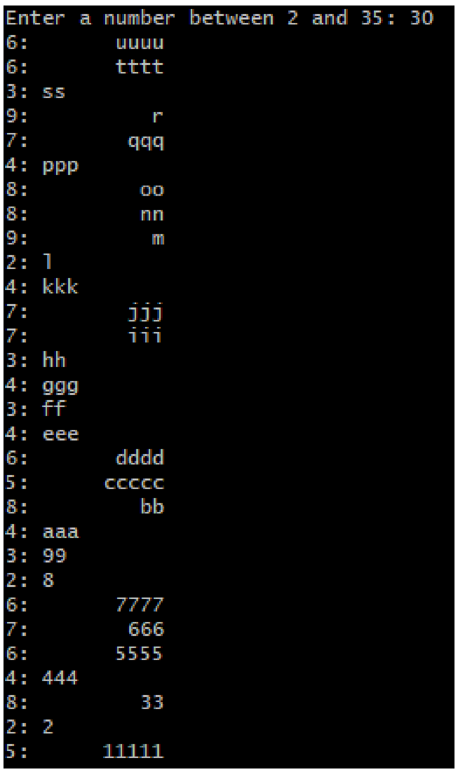
   (a) Modify "`l6_numbers.c`" to increase the upper value of input from 9 to 35. The modified program produces each line in the same way as the original program does. While "`l6_numbers.c`" prints line numbers in a decreasing order, the modify version prints in an increasing order. In addition, when a line number is bigger than 9, a lowercase letter is used for print out in the alphabetic order. Examples of running the modified program are shown below.



```
Enter a number between 2 and 35: 8
4: 888
3: 77
4: 666
8:         55
7:        444
3: 33
5:      22222
8:         11
```



```
Enter a number between 2 and 35: 15
9:          f
9:          e
8:         dd
4: ccc
4: bbb
5:      aaaaa
5:      99999
9:          8
3: 77
2: 6
4: 555
7:        444
2: 3
4: 222
5:      11111
```



```
Enter a number between 2 and 35: 30
6:       uuuu
6:       tttt
3: ss
9:          r
7:        qqq
4: ppp
8:         oo
8:         nn
9:          m
2: l
4: kkk
7:        jjj
7:        iii
3: hh
4: ggg
3: ff
4: eee
6:       dddd
5:      ccccc
8:         bb
4: aaa
3: 99
2: 8
6:       7777
7:        666
6:       5555
4: 444
8:         33
2: 2
5:      11111
```

   (b) For simplicity, it is assumed that the user input is always in range.

   (c) Save your program to a file named "`l7_characters.c`", and submit it online.

Evaluation: All online submissions must be completed before due time, which will be kept on record. In addition, every student is required to show/demonstrate his/her complete exercises to a GA/TA at the end of this lab, or at the beginning of the next lab after completing online submission. The demonstration includes showing the submitted C codes, compiling the C programs, and trying out the C programs with different input values. The maximum marks for this lab is 15, with 10 for the lab work (submission and demonstration) and 5 for lab attendance.