

# C programming problems

Here are few problems to try out your C skills on.

1. Produce a multiplication table. Top left hand corner will show 1x1 and bottom right shows 12x12, as below.

1	2	3	4	5	6	7	8	9	10	11	12
2	4	6	8	10	12	14	16	18	20	22	24
3	6	9	12	15	18	21	24	27	30	33	36
4	8	12	16	20	24	28	32	36	40	44	48
5	10	15	20	25	30	35	40	45	50	55	60
6	12	18	24	30	36	42	48	54	60	66	72
7	14	21	28	35	42	49	56	63	70	77	84
8	16	24	32	40	48	56	64	72	80	88	96
9	18	27	36	45	54	63	72	81	90	99	108
10	20	30	40	50	60	70	80	90	100	110	120
11	22	33	44	55	66	77	88	99	110	121	132
12	24	36	48	60	72	84	96	108	120	132	144

2. Produce an Ascii conversion table.
3. Convert unix files to DOS format.  
Unix records are terminated with x'0a'.  
DOS records are terminated with x'0d0a'.
4. 'for' problem. Count from 1 to 32 and list the range of unsigned integer numbers that can be stored in each collection of bits.  
I.E. 8 bits can hold the range 0 to 255.
5. Random numbers. Produce a sequence of six random numbers Between 1 and 49. These can then be used to play the National lottery (in England). The Odds on getting the right numbers are 13,983,816 to 1 – GOOD LUCK SUCKER.

6. Take the previous 'lotto' program and add code so duplicate numbers are filtered out and the results are sorted (use [qsort](#)).
7. Here is a problem you can solve with an integer array. Generate the Fibonacci sequence. Starting with 0, 1 add them up then take the result and add it to the last number and repeat.

Answer should look like:

0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181 . . .

8. Functions - write a program that calls `bignum()` and `littlenum()` which determine, respectively, the highest and the smallest of a sequence of numbers read in from the terminal.

9. Structures - Use structures to add two fractions together.

10. Write a calendar program.

11. Print the name of the program that's executing.

12. Produce a program that will display a file in hex formats as below.

Most versions of Unix have 'od' but this dump format is more useful.

```

20 2A 20 68 65 78 5F 63 68 61 72 28 63 68 61 72      * hex_char(char
20 2A 70 6F 73 69 74 69 6F 6E 2C 20 63 68 61 72    *position, char
20 63 29 0A 20 20 20 7B 0A 20 20 20 73 70 72 69     c).  {  spr
6E 74 66 28 70 6F 73 69 74 69 6F 6E 2C 20 22 25   ntf(position, "%
30 32 58 20 22 2C 20 63 29 3B 20 0A 0A 09 09 09   02X ", c); .....

```

13. Print yesterdays date. You can use the ANSI standard functions 'time' and 'ctime'. An example of the O/P from my program is

Fri Mar 17 18:46:47 1995

**14.** Display a byte in binary format. For example:

**55 == 00110111**

Things like `<<` `<<=` & can be used to solve this problem.

**Martin Leslie**