



Lab 5: Random Card Hands

Description:

Write a program that reads lines of input containing two integers, **handCount** and **cardCount**. For each line, your program must:

- 1) Generate a randomly ordered, 52-card deck of cards where each card is unique, has one of 13 ranks (A, 2, 3, 4, 5, 6, 7, 8, 9, X, J, Q, K) and one of 4 suits (♣, ♦, ♥, ♠).
- 2) Output **handCount** sets of **cardCount** cards where no card is duplicated, and each of the 52 cards has equal chance of being printed. This is called *uniform sampling without replacement*. Output each card as a pair of two characters: the first character for the rank and the second (C, D, H, or S) for the suit. Separate cards with a comma. Separate each hand with 5 spaces.
- 3) Each hand must be sorted first by rank (A, 2, 3, 4, 5, 6, 7, 8, 9, X, J, Q, K), then by suit (C, D, H, or S).

handCount must be an integer from 0 through 5.

cardCount must be an integer from 0 through 5.

You may assume that your input contains no errors.

Examples:

Input	Possible Output		
3 2	AC, AD	2H, 5C	3C, JD
3 2	8S, XD	4S, 5D	QC, QD
2 3	3D, JD, KH	6D, XH, XS	2S, 3C, 5S
2 5	AC, AD, AH, AS, 2H	4S, 5S, 6S, 7D, 7S	
3 5	5C, 5D, 9S, JS, KH	JS, QH, QS, KD, KS	5H, 7D, 7H, 7S, XC
0 5			
4 0			
1 1	JD		

Grading Rubric (total of 20 points)

[-2 points]: Program compiles with warnings on `cs.unm.edu` using `/usr/bin/gcc` with no options.

[-5 points]: Code does not follow the CS-241 standard.

[1 point]: The program is attached in Blackboard Learn as a single .c file:

`CS-241.randomHands.yourfirstname.yourlastname.c`

[6 points]: The program outputs correct format and correct number of cards. Each time the program is run the output is different.

[7 points]: Each card had equal chance and no repeats.

[6 points]: Each hand is correctly sorted.

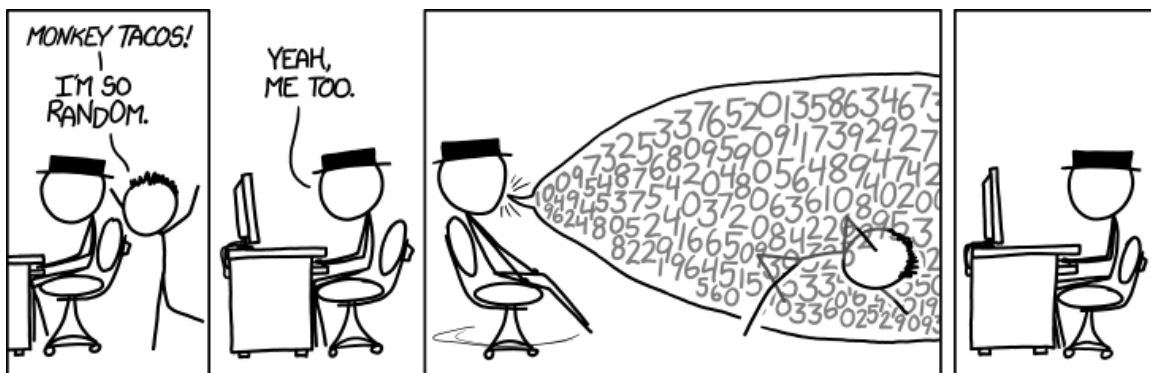
Hints:

I suggest representing, storing and sorting each card as a single integer: 0 through 51.

I suggest the only time you convert the integer version to a pair of characters is in a helper function for printing. The helper function could take an int and a char array of length 3 as arguments (rank, suit and `'\0'`). Note: a function cannot return an array because (unless you use malloc) the array would go on the stack and get cleared when the function returns.

If you do represent each card as a single int, n , then $\text{rank} = n/4$ and $\text{suit} = n \% 4$ (why?). Once you have rank and suit as numbers 0 through 12 and 0 through 3, you could use a long series of **if** statements to convert to the required characters, but much simpler would be to use rank and suit as indexes to character arrays:

```
char rankChar[] = "A23456789XJQK";  
char suitChar[] = "CDHS";  
// soooo simple, it is beautiful 😊
```



xkcd.com