Obfuscated Code


C code to print all 12 verses of The 12 Days of Christmas:

```c
#include <stdio.h> main(t,_,a){return!0<t?t<3?main(-79,-13,a
+main(-87,1-_ , main(-86,0,a+1)+a)):1,t<_?main(t+1,_ ,a):
3,main(-94,-27+t,a)&&t==2?_<13? main(2, _+1,"%s %d %d\n") :9:16:t<0?t<-72?
main(_ ,t , "@n'+,##/{*}w+w/w#cdnr+,{}r/*de}+,/*{+,*%/w{+,/w#q#n+,,#{l,+,/n{n
+,/+#n+,,/\ ;#q#n+,/+k#;*+,,/'r :"d*3,}{w+K w'K:'+}e#';dq#'l \ q#'d'K#/ +k#;q#'r}eKK#}w'r}eKK{nl}'/#;#q#n'}{#}w')(){}{nl}'/+#n';d}rw' i;# \ ){}nl!/ n{nl}; r{#w'r nc{nl}'/#{1,+'K {rw' iK{}[{nl]}/w#q#n'wk nw' \ iwK{KK}[nl]!/ w{%l##w' # i; :{}nl']/*{q#'ld;'}{nlw!/*de}'c \ ;{nl}'-{rw}'/,+}##'*
#nc,',#nw}'+/kd'+e}++;#rdq#w! nr'/ ') )}{rl'{}{n' '}) # \ }'++)##(!/")) :t<-50?
_==*a?putchar(31[a]):main(-65,_,a+1):main(*a=''/)+t,_,a+1) :0<t?
main(2,2,"%s"):a=''/ ||main(0,main(-61,*a, "!ek;dc i@bK'(q)−[w]*%n+r3#1,}{:
\nuwloca-0;m .vpbks,fxntdCeghiry"),a+1);}
```
• Homework for Thursday
  – Upload a question, and answer, to Moodle that’s not in the book but Matt could ask on the final.

• Thursday class, 12/6: Review session
  – Come with questions!
  – Don’t forget about practice problems on website

• Next Monday, 12/10: 10:30-noon, office hour

• Final Exam
  – Thursday, 12/13, noon-3pm
  – Monday, 12/17, noon-3pm
• Pseudocode on test

3. What color light does a ceiling light emit? (5 points)

4. Write down Planck’s equation. (10 points)

5. Why were you able to observe individual wavelengths? (10 points) Spectroscopy is...
• 15% Exam 1
• 15% Exam 2
• 20% Final Exam
• 15% Labs
• 10% Project 1
• 10% Project 2
• 10% (Homework / Quizzes / Reading responses)
• 5% Class participation
2d arrays (list of lists)
for i in range(ROWS):
    for j in range(COLS):
        if random() < 0.1:
            board[i][j] = 1
• Write pseudocode for a function `showWalks(n, m)` that simulates a 2D random walk `n` times for `m` steps, on a 10x10 board. Assume that the simulation starts with the robot at 5, 5 and there’s an equal chance of it moving N/S/E/W on each time step. After the `n` simulations, print out how many times the agent ended up in each of the 100 locations.

• Use top-down programming. Anything you don’t know how to do, or don’t have time to do, assume your labmate will write it, based on your detailed description.
Recursion – not on test

def factLoop(n):
    answer = 1
    for i in range(1,n+1):
        answer = answer * i
    return answer

def factRecursion(n):
    if n <= 1:
        return 1
    else:
        return n * factRecursion(n-1)