$$-9 = \_\_ \cdot 11 + \_\_$$

$$-9 = (-1) \cdot 11 + 2$$

$$GCD(100, 15)$$

$$100 = \underline{6} \cdot 15 + \underline{10}$$

$$15 = \underline{1} \cdot 10 + 5$$

$$10 = \underline{2} \cdot 5 + \underline{0}$$

$$2^{130} \pmod 7$$

$$2^1 \equiv 2 \pmod 7$$

$$2^2 \equiv 2^2 \equiv 4$$

$$2^4 \equiv 4^2 \equiv 16 \equiv 2$$

$$2^8 \equiv 2^2 \equiv 4$$

$$2^{16} \equiv 4^2 \equiv 16 \equiv 2$$

$$2^{32} \equiv 4$$

$$2^{64} \equiv 2$$

$$2^{128} \equiv 4$$

$$130 = 128 + 2$$

$$2^{130} \equiv 2^{(128+2)}$$

$$\equiv 2^{128} \cdot 2^2$$

$$\equiv 4 \cdot 4$$

$$\equiv 16$$

$$\equiv 2 \pmod 7$$

$2^{6002} \pmod 7$

$2^6 \equiv 1 \pmod 7$ by F.L.T.

$2^{6000} = 2^{6 \cdot 1000} = (2^6)^{1000} \equiv 1^{1000} \equiv 1 \pmod 7$

$2^2 \equiv 4 \pmod 7$

$2^{6002} = 2^{6000+2} = (2^{6000}) \cdot (2^2) \equiv 1 \cdot 4 \equiv 4$

CSE 215 Programming Assignment 2

September 21, 2025

For your second programming assignment, let's play with prime numbers! More specifically, we're going to look at *factors* of numbers.

To find the factors of a number, you can simply divide the number by consecutive integers, and see which integers divide it evenly. But rather than divide, you can use the modulus function (%):

n%f returns the remainder when n is divided by f. So if n%f==0, you know that f is a factor of n

So let's write a program that accepts a number (n) from the user, analyzes it, and reports the following facts about the number:

1. a list of all factors between 1 and n-1;
2. a statement about whether the number is prime or composite (not prime);
3. the *sum* of all the factors from 1 to n-1; and
4. a statement as to the *mood* of the number.

The mood is found as follows (let's suppose s=the sum of all factors from 1 to n-1):

- if $s < n$, the number is "sad";
- if s>n, the number is "happy";
- if s=n, the number is "perfect."

So for item 4. above, your code should say the number is sad, happy or perfect.

Requirements

Besides the above behavior, there are the following additional requirements:

- your program must be written in C, and all code must be written by

$1 = 1 = 1^2$

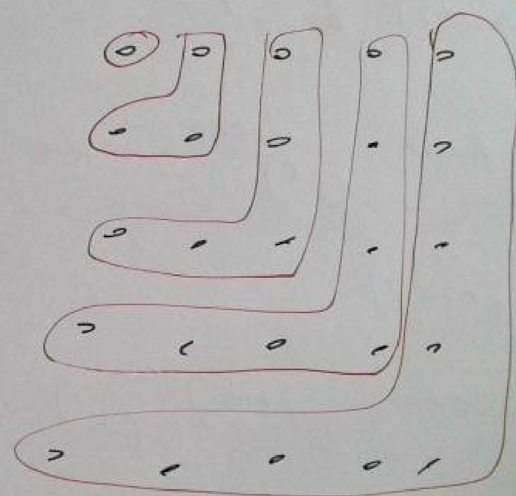
$1+3 = 4 = 2^2$

$1+3+5 = 9 = 3^2$

$\vdots$

$1+3+5+7 = 16$

$1+3+5+7+9 = 25$



Prove $1+3+\cdots+(2n-1)=n^2 \quad \forall n \geq 1$

Proof: Let $P(n)$ be the prop: $1+3+\cdots+(2n-1)=n^2 \quad (n \geq 1)$

I BASE CASE. $P(1)$ says $1=1^2$ <u>True</u>

II Inductive step: prove $P(n) \Rightarrow P(n+1) \quad (n \geq 1)$

proof:

1. $P(n)$ Hypothesis
2. $1+3+\cdots+(2n-1)=n^2$ Def of $P(n)$, 1
3. $\underbrace{1+3+\cdots+(2n-1)}_{n^2}+(2n+1)=$

$n^2 + (2n+1)$ Subs. 2

$= n^2+2n+1 = (n+1)^2$ alg.

4. $P(n+1)$ Def of $P(n+1)$, 3.

$\therefore P(n) \to P(n+1)$

$\therefore P(n) \ \forall n \geq 1$ M.I.

QED

<u>IGNORE</u>

WTS $P(n+1)$

$1+3+\cdots+(2n-1)+(2n+1)=(n+1)^2$

Prove $2^n > 1 \quad \forall n \geq 1$

Let $p(n)$ be the prop $2^n > 1 \quad (n \geq 1)$

Prove $p(n) \ \forall n \geq 1$

Base Case: $p(1)$ says $2^1 > 1$ True

Prove $p(n) \to p(n+1)$

1. $p(n)$ Hyp
2. $2^n > 1$ def of $p(n)$
3. $\underline{2^{n+1}} = 2 \cdot 2^n > 2 \cdot 1 = 2 > \underline{1}$ alg.
4. $p(n+1)$ Def of $p(n+1)$, 3

$\therefore p(n) \to p(n+1)$

$\therefore p(n) \ \forall n \geq 1$ M.I.

QED

$2^3 > 1$

$2^4 = 2 \cdot 2^3 > 2 \cdot 1 = 2 > 1$

$\underline{2}^5 = 2 \cdot 2^4 > 2 \cdot 1 = 2 > \underline{1}$

Prove $n < 2^n \ \forall n \geq 1$

Let $P(n)$ be the prop $n < 2^n$

Prove $P(n) \ \forall n \geq 1$

Base case: $P(1)$ says $1 < 2^1$ True

Prove $P(n) \to P(n+1)$

1. $P(n)$ Hyp
2. $n < 2^n$ Def of $P(n)$
3. $2^{(n+1)} = 2 \cdot 2^n = 2^n + 2^n > n + n > n+1$ alg. inequality 2.
4. $P(n+1)$ Def of $P(n+1)$, 3.

⋮

IGNORE

WTS

$n+1 < 2^{(n+1)}$

Prove $n < 2^n \quad \forall n \geq 1$

Let $p(n)$ be the prop $n < 2^n$

Prove $p(n) \ \forall n \geq 1$

Base case: $p(1)$ says $1 < 2^1$ True

Prove $p(n) \rightarrow p(n+1)$

1. $p(n)$ Hyp
2. $n < 2^n$ Def of $p(n)$
3. $2^{(n+1)} = 2 \cdot 2^n = 2^n + 2^n > n + n > n+1$ alg. inequality 2.
4. $p(n+1)$ Def of $p(n+1)$, 3.

⋮

IGNORE

WTS

$n+1 < 2^{(n+1)}$

$$\begin{array}{l} P(1) \\ P(n) \to P(n+1) \\ \hline P(n) \quad \forall n \geq 1 \end{array}$$

$P(1)$

$P(n) \to P(n+1)$

$P(n) \quad \forall n \geq 1$

Prove $|\mathcal{P}(S)| = 2^{|S|}$, $|S| \geq 1$ (S finite)

ex $\mathcal{P}(\{a,b,c\}) = \{\{a\}, \{b\}, \{c\},$
$\{a,b\}, \{a,c\}, \{a,b,c\},$
$\{b,c\}, \emptyset\}$

$8 = 2^3$

Let $p(n)$ be the prop: if $|S| = n$ then $|\mathcal{P}(S)| = 2^n$

~~Let~~ Prove $p(n)$ $\forall n \geq 1$

BASE CASE: $P(1)$ says if $S = \{x\}$ then $|\mathcal{P}(S)| = 2^1 = 2$

$\mathcal{P}(S) = \{ S, \phi\}$ True

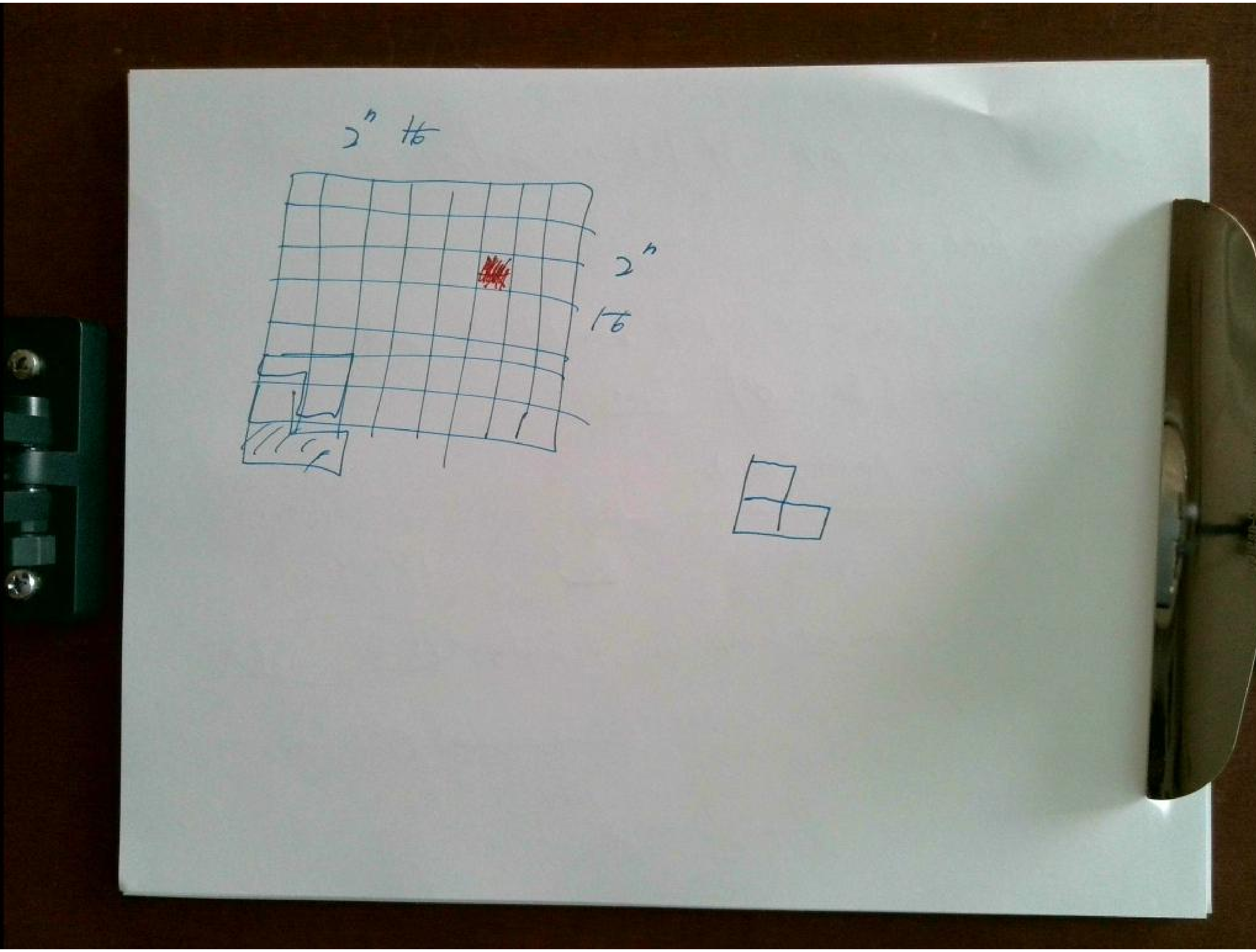
Inductive Step: Prove $p(n) \rightarrow p(n+1)$

Let $S = \{s_1, s_2, s_3, \dots, s_n, s_{n+1}\}$ $|S| = n+1$

Subsets of S: if it does not include s_{n+1}, it's a subset of $\{s_1, s_2, \dots, s_n\}$

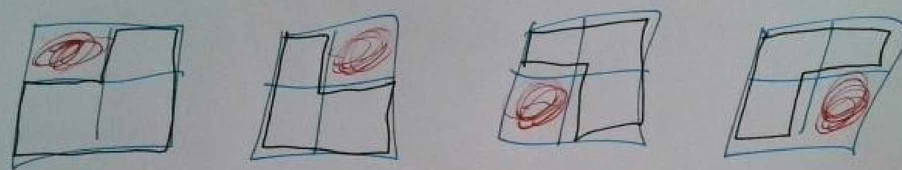
If it does include s_{n+1}, it's a subset of $\{s_1, s_2, \dots, s_n\}$ plus s_{n+1}

$$2^n + 2^n = 2^{n+1}$$


2^n 16
2^n
16

BASE CASE

P(1) says you can tile a $2^1 \times 2^1 = 2 \times 2$ room (with one missing square)



P(1) is true

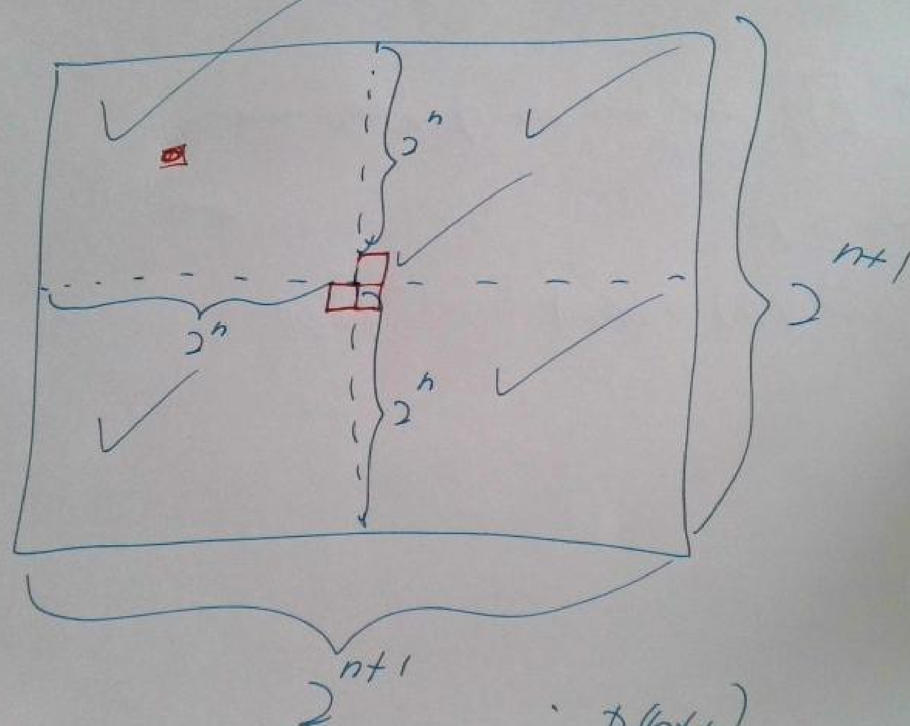
Let $p(n)$ be the prop: you can tile* a $2^n \times 2^n$ room

* using funny tiles:  and with one square uncovered.

Prove $p(n)$ $\forall n \geq 1$.

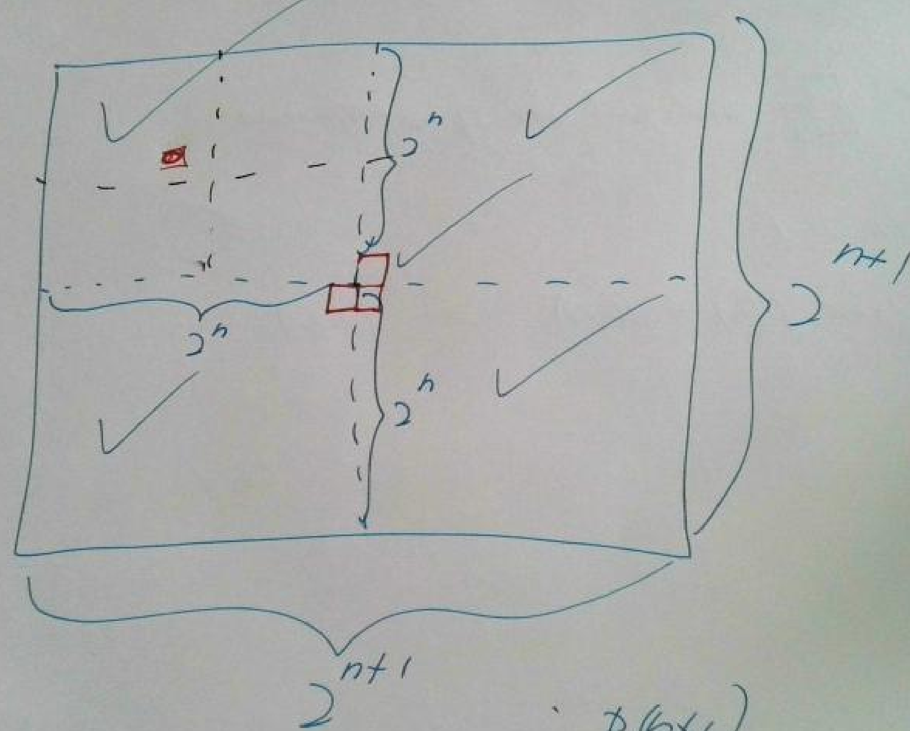
Inductive step: prove $p(n) \to p(n+1)$ $(n \geq 1)$

given $P(n)$ is true. Prove $P(n+1)$



$\therefore P(n+1)$

given $P(n)$ is true. Prove $P(n+1)$



$\therefore P(n+1)$

RECURSION

Ex: find the length of a string.

if string is empty, return(0)

return 1 + lengthof(string w/ one char. removed)

POSTAGE STAMPS

given an unlimited supply of 3¢ & 7¢ stamps,

Prove that you can make exact postage for any amount ≥ 21¢

$21 = 7+7+7$

$22 = 3+3+3+3+3+7$

$23 = 7+7+3+3+3$

Let $P(n)$ be the prop: you can make n¢ from 3's & 7's

prove $P(n)$ $\forall n \geq 21$

BASE CASE: $P(21)$ ... True

Prove $P(n) \Rightarrow P(n+1)$

2 CASES:

1) If you have at least 2 3¢ stamps in the solution for n¢
   remove 3¢ 3¢, add 7¢

2) there are only one 3¢ or no 3¢ stamps in the solution for n¢
   there are at least two 7¢ stamps
   remove 7¢ 7¢, add 3¢ 3¢ 3¢ 3¢ 3¢

Mon 03 Nov 2025, 13:47:50 towers : vi — Konsole

towers : vi — Konsole

File Edit View Bookmarks Plugins Settings Help

towers : vi towers : jam22

```
  }

// solve for total-1
  solve(total-1,three,seven);

// If there are *at least* two 3-cent stamps,
// we can replace one with a 7-cent stamp
// (total postage increases by 1).

  if (*three >= 2){
    (*three)=(*three)-2;++(*seven);return;
  }

// Otherwise, there's only one 3 cent stamp,
// so the 7 cents stamps add up to at least 17-3=14,
// so there are at least *two* 7 cent stamps.
// Replace two 7's with five 3's
// (total postage increases by 1).

  (*seven)=(*seven)-2;
  (*three)=(*three)+5;
}
                                                    34,1          Bot
```

POSTAGE STAMPS

given an unlimited suppbot 3¢ & 7¢ stamps,

Prove that you can make exact postage for any amount ≥ 21¢

$21 = 7+7+7$

$22 = 3+3+3+3+3+7$

$23 = 7+7+3+3+3$

Let $P(n)$ be the prop: you can make n¢ from 3's & 7's

prove $P(n)$ $\forall n \geq 21$

BASE CASE: $P(21)$ ... True

Prove $P(n) \Rightarrow P(n+1)$

2 CASES:

1) If you have at least 2 3¢ stamps in the solution for n¢
   remove 3¢ 3¢, add 7¢

2) there are only one 3¢ or no 3¢ stamps in the solution for n¢
   there are at least two 7¢ stamps
   remove 7¢ 7¢, add 3¢ 3¢ 3¢ 3¢ 3¢