Intro to Computer Science & Programming *for Adults*

Richard White
rwhite@polytechnic.org

Polytechnic School
Jan-Feb, 2025

Welcome Back! All about AI

Week 2 – The Internet, Webpages, & Cybersecurity

1. How the Internet Works
2. How web pages work
3. Cybersecurity: Threat models, vectors, backups
4. “Have I Been Pwned?” [Activity]
5. Password Managers [Demo]
6. Break
7. Programming: Random numbers (*pickcode.io*)
8. Making decisions: `if-else`
9. Programming assignments: Magic 8 Ball, Guess A Number

Week 3 – AI, Machine Learning, & Classifiers

1. What even *is* AI anyway?
2. Machine Learning
3. Large Language Models (LLMs)
4. Classifiers [Activity]
5. Break
6. Programming: loops (*pickcode.io*)
7. infinite loops
8. Number Guessing Game

Week 4-TBD

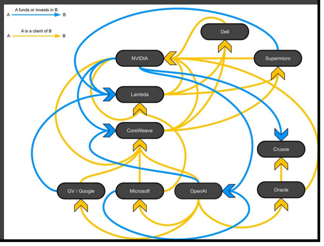
- Creating a personal webpage
- Social media, social networks
- Algorithmic Justice (FLOCK camera, Facial recognition, bias)
- Gaming
- Enshittification
- Robots
- Open conversation

What even *is* AI, anyway?

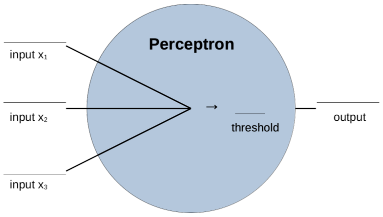
- Historical Artificial Intelligence: a computer that exhibits human-level intelligence. (The Turing Test)
- Artificial General Intelligence (AGI): a computer that matches or surpasses human cognitive abilities
- Machine Learning: uses a trained model to identify patterns in data
- Generative AI: a type of machine learning in which the patterns are used to generate new data (visual, aural, text)
- Large Language Models (LLMs): generative AI models that focus on text, often supplemented with auxiliary tools

The scope of conversation

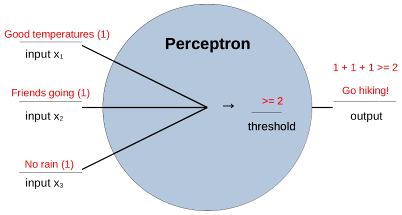
- AI as a branch of computer science
- AI as a tool for coders, artists, students, people
- ~~AI as hyped by tech bros~~
- ~~AI as an economic / financial market concern~~
- ~~AI as a resource issue (power, water)~~



Machine Learning - Perceptron



Machine Learning - Perceptron



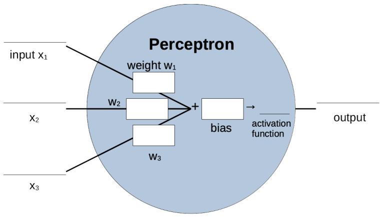
Predicted Examples

Good temperatures	Friends going	No rain	Go Hiking
1	1	1	1
1	1	0	1
1	0	1	1
0	1	1	1
0	0	1	0
0	1	0	0
0	0	0	0

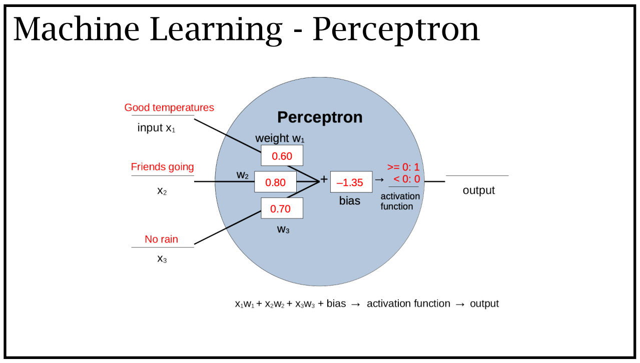
"Actual Data" for me hiking

Good temperatures	Friends going	No rain	Go Hiking
1	1	1	1
1	1	0	0
1	0	1	1
0	1	1	1
0	0	1	0
0	1	0	1
0	0	0	0

Machine Learning - Perceptron

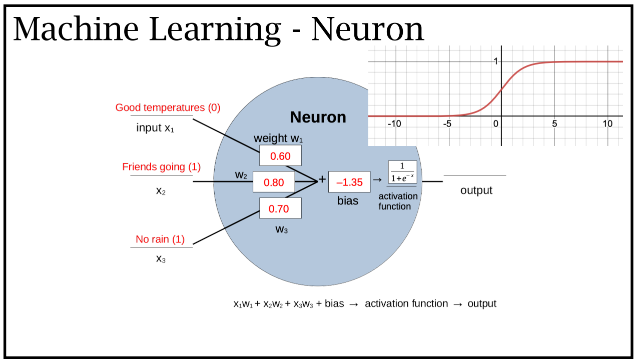


$x_1w_1 + x_2w_2 + x_3w_3$ + bias → activation function → output



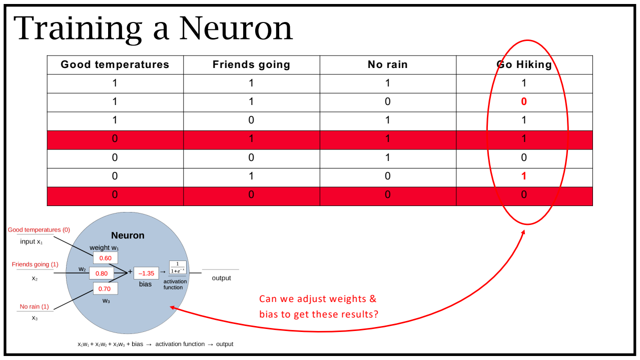
Machine Learning - Perceptron

$x_1w_1 + x_2w_2 + x_3w_3 + bias$ → activation function → output



Machine Learning - Neuron

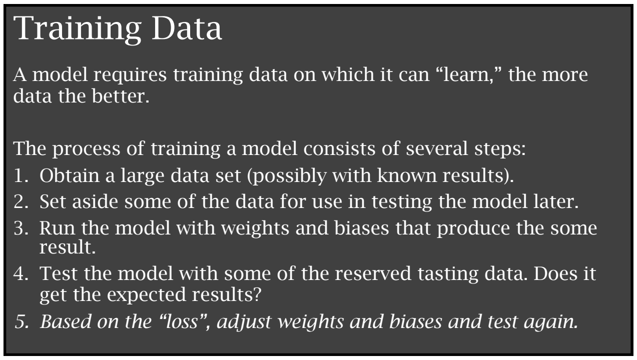
$x_1w_1 + x_2w_2 + x_3w_3 + bias$ → activation function → output



Training a Neuron

Good temperatures	Friends going	No rain	Go Hiking
1	1	1	1
1	1	0	0
1	0	1	1
0	1	1	1
0	0	1	0
0	1	0	1
0	0	0	0

Can we adjust weights & bias to get these results?

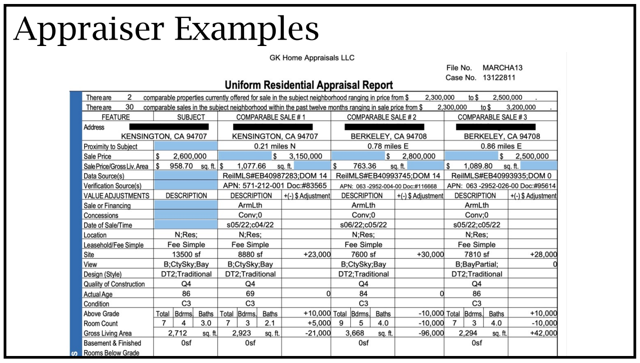


Training Data

A model requires training data on which it can "learn," the more data the better.

The process of training a model consists of several steps:

1. Obtain a large data set (possibly with known results).
2. Set aside some of the data for use in testing the model later.
3. Run the model with weights and biases that produce the some result.
4. Test the model with some of the reserved tasting data. Does it get the expected results?
5. *Based on the "loss", adjust weights and biases and test again.*



Appraiser Examples

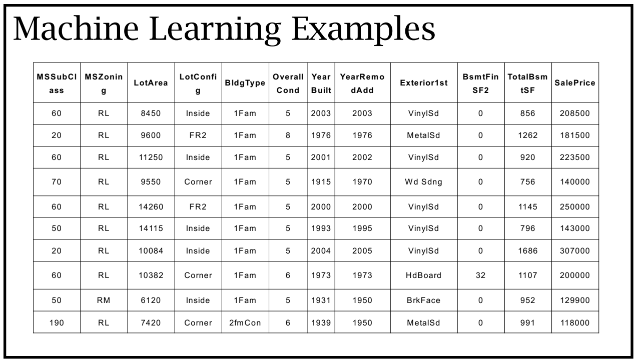
GK Home Appraisals LLC

File No. MARCHA13
Case No. 13122811

Uniform Residential Appraisal Report

There are 2 comparable properties currently offered for sale in the subject neighborhood ranging in price from $ 2,300,000 to $ 2,500,000 .
There are 30 comparable sales in the subject neighborhood within the past twelve months ranging in sale price from $ 2,300,000 to $ 3,200,000 .

FEATURE	SUBJECT	COMPARABLE SALE # 1		COMPARABLE SALE # 2		COMPARABLE SALE # 3	
Address	[redacted] KENSINGTON, CA 94707	[redacted] KENSINGTON, CA 94707		[redacted] BERKELEY, CA 94708		[redacted] BERKELEY, CA 94708	
Proximity to Subject		0.21 miles N		0.78 miles E		0.86 miles E	
Sale Price	$ 2,600,000		$ 3,150,000		$ 2,800,000		$ 2,500,000
Sale Price/Gross Liv. Area	$ 958.70 sq. ft.	$ 1,077.66 sq. ft.		$ 763.36 sq. ft.		$ 1,089.80 sq. ft.	
Data Source(s)		ReilMLS#EB40987283;DOM 14		ReilMLS#EB40993745;DOM 14		ReilMLS#EB40993935;DOM 0	
Verification Source(s)		APN: 571-212-001 Doc:#83565		APN: 063 -2952-004-00 Doc:#116668		APN: 063 -2952-026-00 Doc:#95614	
VALUE ADJUSTMENTS	DESCRIPTION	DESCRIPTION	+(-) $ Adjustment	DESCRIPTION	+(-) $ Adjustment	DESCRIPTION	+(-) $ Adjustment
Sale or Financing		ArmLth		ArmLth		ArmLth	
Concessions		Conv;0		Conv;0		Conv;0	
Date of Sale/Time		s05/22;c04/22		s06/22;c05/22		s05/22;c05/22	
Location	N;Res;	N;Res;		N;Res;		N;Res;	
Leasehold/Fee Simple	Fee Simple	Fee Simple		Fee Simple		Fee Simple	
Site	13500 sf	8880 sf	+23,000	7600 sf	+30,000	7810 sf	+28,000
View	B;CtySky;Bay	B;CtySky;Bay		B;CtySky;Bay		B;BayPartial;	0
Design (Style)	DT2;Traditional	DT2;Traditional		DT2;Traditional		DT2;Traditional	
Quality of Construction	Q4	Q4		Q4		Q4	
Actual Age	86	69	0	84	0	86	
Condition	C3	C3		C3		C3	
Above Grade	Total Bdrms. Baths	Total Bdrms. Baths	+10,000	Total Bdrms. Baths	-10,000	Total Bdrms. Baths	+10,000
Room Count	7 4 3.0	7 3 2.1	+5,000	9 5 4.0	-10,000	7 3 4.0	-10,000
Gross Living Area	2,712 sq. ft.	2,923 sq. ft.	-21,000	3,668 sq. ft.	-96,000	2,294 sq. ft.	+42,000
Basement & Finished Rooms Below Grade	0sf	0sf		0sf		0sf	



Machine Learning Examples

MSSubClass	MSZoning	LotArea	LotConfig	BldgType	Overall Cond	Year Built	YearRemodAdd	Exterior1st	BsmtFinSF2	TotalBsmtSF	SalePrice
60	RL	8450	Inside	1Fam	5	2003	2003	VinylSd	0	856	208500
20	RL	9600	FR2	1Fam	8	1976	1976	MetalSd	0	1262	181500
60	RL	11250	Inside	1Fam	5	2001	2002	VinylSd	0	920	223500
70	RL	9550	Corner	1Fam	5	1915	1970	Wd Sdng	0	756	140000
60	RL	14260	FR2	1Fam	5	2000	2000	VinylSd	0	1145	250000
50	RL	14115	Inside	1Fam	5	1993	1995	VinylSd	0	796	143000
20	RL	10084	Inside	1Fam	5	2004	2005	VinylSd	0	1686	307000
60	RL	10382	Corner	1Fam	6	1973	1973	HdBoard	32	1107	200000
50	RM	6120	Inside	1Fam	5	1931	1950	BrkFace	0	952	129900
190	RL	7420	Corner	2fmCon	6	1939	1950	MetalSd	0	991	118000

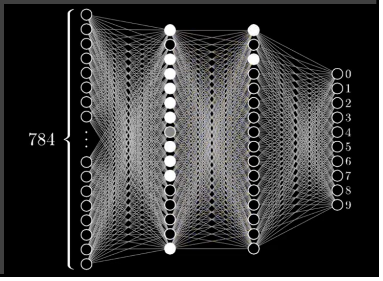
Machine Learning

In supervised learning, large training sets of data are assembled: examples, made up of features, each of which have a result, or label.

This data is provided to an AI system that develops a model based on the data and an iterative process. This *model* is a set of numbers (weights and biases among them) that attempt to describe the relationship between the features in an example and their labels.

A Layer of Neurons

Our neuron example had 3 inputs, a single neuron, and a single output. Because it's easy to create more complex structures, most Machine Learning algorithms use many more inputs, and multiple *layers* of neurons to better develop the *weights* and *biases* that make up a good model.



Train a Classifier [Activity]

Go to https://teachablemachine.withgoogle.com and give the website access to your camera and/or microphone.

Follow the instructions and see if you can train the model to recognize the difference between two or three things.

Soda versus chips?

Red playing cards vs. black playing cards?

Cash vs. coins?

Red versus green?

Generative AI

“Generative AI” consists of models trained on data, with the resulting models used to generate new data.

- Generated text (Large Language Models like ChatGPT, Gemini)
- Generated text (coding-focused models like Claude)
- Generated images
- Generated audio
- Generated video

Large Language Models (LLMs)

LLMs are based on collecting vast amounts of text (primarily from the Internet and books) and training models on that text.

The resulting models, when crafting a reply to a response, assemble words based on the probability of those words (”tokens”) occurring together ”in real life.”

The model isn't aware of what it's doing, and doesn't understand what it's doing. The words it assembles, however, are often so realistically chosen that it *seems* as if it know what it's saying.

It doesn't.

LLM Python-based query [Demo]

Most LLMs provide developers with access to the models via an *Application Programming Interface* (API).

Each time you interact with the model, you're charged some amount for the transaction.

Here's what that looks like.

OpenAI API call - Python

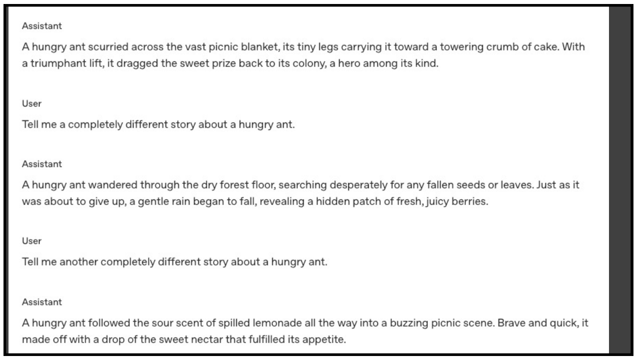
```
from openai import OpenAI

client = OpenAI()
response = client.responses.create(
    model='gpt-5-nano',
    input='Write a three-sentence bedtime story with a
scary, suspenseful storyline, but without any violence
or blood.'
)
print(response.output_text)
```

What else do we know about LLMs?

- They can be enormously useful, particularly when provided with a large amount of context.
- They are probabilistic. The same prompts given to the same LLM will produce different results.
- It's not *just* random words. The most recent LLMs are able to build a profile of you over time, and use those in responding to your queries.
- They won't ever become AGI. There are plenty of other reasons to be worried about them, however.
- They are algorithms that don't have sentience, creativity, or emotion.

LLMs on the back end

Exploring LLMs, System prompts, Temperature



Assistant

A hungry ant scurried across the vast picnic blanket, its tiny legs carrying it toward a towering crumb of cake. With a triumphant lift, it dragged the sweet prize back to its colony, a hero among its kind.

User

Tell me a completely different story about a hungry ant.

Assistant

A hungry ant wandered through the dry forest floor, searching desperately for any fallen seeds or leaves. Just as it was about to give up, a gentle rain began to fall, revealing a hidden patch of fresh, juicy berries.

User

Tell me another completely different story about a hungry ant.

Assistant

A hungry ant followed the sour scent of spilled lemonade all the way into a buzzing picnic scene. Brave and quick, it made off with a drop of the sweet nectar that fulfilled its appetite.

Break

Back in 10 minutes?

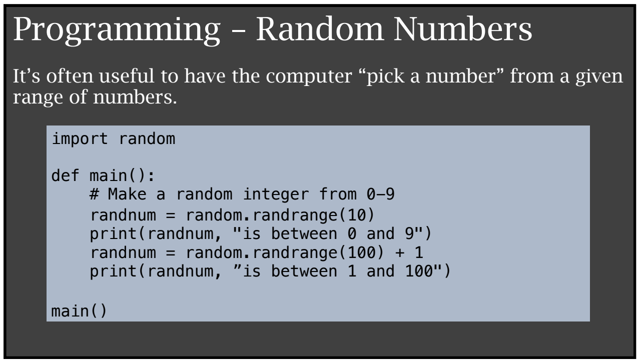
Programming: Loops

Computers have output and input: `print()`, `input()`

Computers can calculate w/ numbers: `age = age + 1`

Computers can make decisions: `if-else` statements

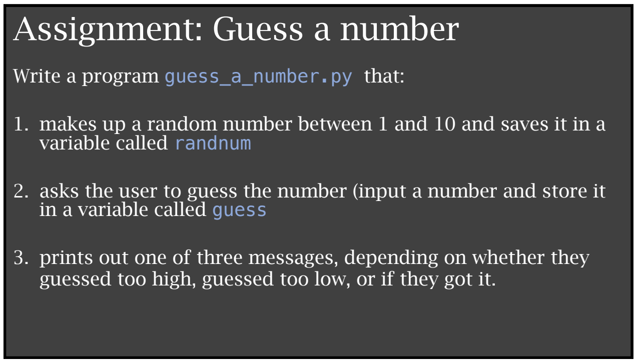
Computers can *loop*, repeating blocks of code: `for` statements



Programming – Random Numbers

It's often useful to have the computer "pick a number" from a given range of numbers.

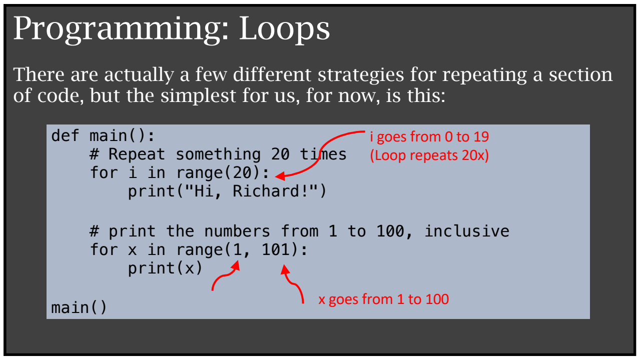
```
import random

def main():
    # Make a random integer from 0-9
    randnum = random.randrange(10)
    print(randnum, "is between 0 and 9")
    randnum = random.randrange(100) + 1
    print(randnum, ”is between 1 and 100")

main()
```



Assignment: Guess a number

Write a program `guess_a_number.py` that:

1. makes up a random number between 1 and 10 and saves it in a variable called `randnum`
2. asks the user to guess the number (input a number and store it in a variable called `guess`
3. prints out one of three messages, depending on whether they guessed too high, guessed too low, or if they got it.



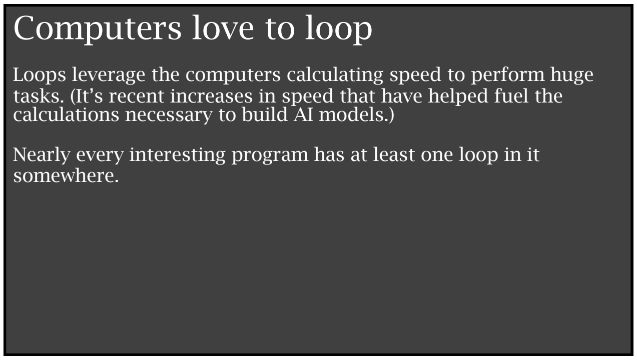
Programming: Loops

There are actually a few different strategies for repeating a section of code, but the simplest for us, for now, is this:

```
def main():
    # Repeat something 20 times
    for i in range(20):
        print("Hi, Richard!")

    # print the numbers from 1 to 100, inclusive
    for x in range(1, 101):
        print(x)

main()
```

i goes from 0 to 19
(Loop repeats 20x)

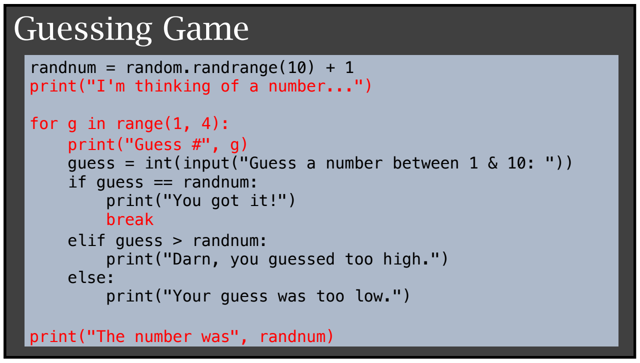
x goes from 1 to 100



Computers love to loop

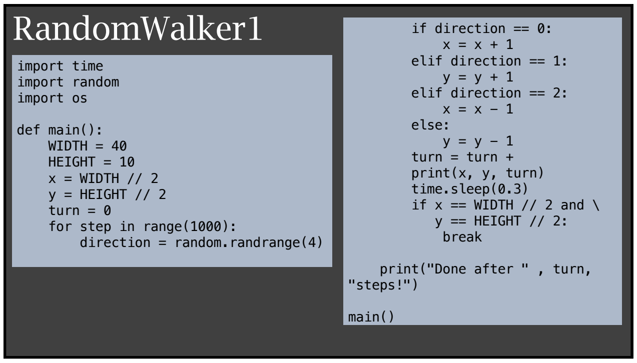
Loops leverage the computers calculating speed to perform huge tasks. (It's recent increases in speed that have helped fuel the calculations necessary to build AI models.)

Nearly every interesting program has at least one loop in it somewhere.



Guessing Game

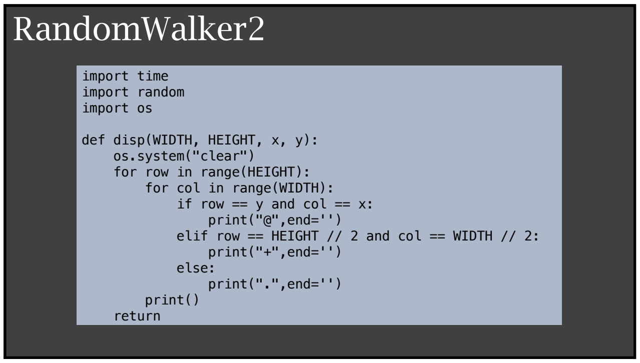
```
randnum = random.randrange(10) + 1
print("I'm thinking of a number...")

for g in range(1, 4):
    print("Guess #", g)
    guess = int(input("Guess a number between 1 & 10: "))
    if guess == randnum:
        print("You got it!")
        break
    elif guess > randnum:
        print("Darn, you guessed too high.")
    else:
        print("Your guess was too low.")

print("The number was", randnum)
```



RandomWalker1

```
import time
import random
import os

def main():
    WIDTH = 40
    HEIGHT = 10
    x = WIDTH // 2
    y = HEIGHT // 2
    turn = 0
    for step in range(1000):
        direction = random.randrange(4)
```

```
        if direction == 0:
            x = x + 1
        elif direction == 1:
            y = y + 1
        elif direction == 2:
            x = x – 1
        else:
            y = y – 1
        turn = turn +
        print(x, y, turn)
        time.sleep(0.3)
        if x == WIDTH // 2 and \
           y == HEIGHT // 2:
            break

    print("Done after " , turn,
"steps!")

main()
```



RandomWalker2

```
import time
import random
import os

def disp(WIDTH, HEIGHT, x, y):
    os.system("clear")
    for row in range(HEIGHT):
        for col in range(WIDTH):
            if row == y and col == x:
                print("@",end='')
            elif row == HEIGHT // 2 and col == WIDTH // 2:
                print("+",end='')
            else:
                print(".",end='')
        print()
    return
```