

Day 1: The tip Calculator

This program uses simple mathematics to split bill between number of people

Program in Action:

Welcome to the tip calculator!

What is the total bill amount?

>>\$:500

How much tip would you like to give?

>>Percent:5

How many people to split the bill?

>>People:10

Each person should pay: \$52.50

Day 2: Treasure Island

This is a simple game that uses if else statements in story and asks user to make this or that decisions and develops story accordingly leading to 1 winning or multiple losing situations.

Story: -

User is in an island searching a way to the Treasure Island. He has options to go left or right if user chooses right, Sonic the Hedgehog gets first to the island and user loses. If he goes left, he reaches a beach from where he sees the island and he can either swim or wait for a ship to pass by. If he swims, he is eaten by a shark and loses if he waits, he gets to island in a ship shortly. after getting off the ship he sees a previously dug site and a cave nearby. If he searches the cave he is eaten by a bear and loses if he digs the site, he gets all the treasure and wins the game.

Program in action:

Welcome to Treasure Island.

Your mission is to find the treasure.

Left or right?

>>Type Right/Left: left

Nice, you made it to the next level!

Your map shows that you need to get to Treasure Island, you can wait to board a ship or swim accross the sea, pick one.

>>Type Swim/Wait: wait

Nice, you made it to the next level, you're pretty good at this!

Now that you've made it to Treasure Island, you can dig or search the cave.

>>Type Dig/Cave: dig

You've found the treasure, you win!

Day 3: Guess the number

Guess the number, it's a simple game where computer randomly chooses number between 1 to 100. User tries to guess the unknown number within certain number of guesses. In every guess the program replies with "too low" or "too high" until user makes correct guess or runs out of guess. You can increase or decrease the difficulty by adding or subtracting number of guess and range of numbers.

Program in Action:

I'm thinking of a number between 1 and 100, try to guess it

Choose a difficulty. Type 'easy' or 'hard': easy

You have 10 guesses left for the number that I'm thinking of.

>>Take you guess: 1

Too low.

You have 9 guesses left for the number that I'm thinking of.

>>Take you guess: 20

Too low.

.....

You have 4 guesses left for the number that I'm thinking of.

>>Take you guess: 85

Too high.

You have 3 guesses left for the number that I'm thinking of.

>>Take you guess: 83

Correct! The answer was 83. Thanks for completing that!

Do you want to play again? Type 'y' if yes and 'n' to quit.

Thank You !

Day 4: Scissors Paper Rock

This is very famous game of Scissors Paper Rock. Program asks user for his choice for S/P/R and computer also randomly chooses one and you know what to do next.

Program in action:

What do you choose? Type 0 for Rock, 1 for Paper or 2 for Scissors

>>0/1/2:0

You chose: Rock

The computer chose: Paper

You lose, Paper wins against rock.

Day 5: Caesar Cipher

The Caesar cipher is an ancient encryption algorithm used by Julius Caesar. It encrypts letters by shifting them over by a certain number of places in the alphabet. We call the length of shift the key. For example, if the key is 3, then A becomes D, B becomes E, C becomes F, and so on. To decrypt the message, you must shift the encrypted letters in the opposite direction. This program lets the user encrypt and decrypt messages according to this algorithm.

Program in action:

Type 'encode' to encrypt, type 'decode' to decrypt:

>>encode

Type your message:

>>Hello World

Type the shift number:

>>2

Here's the encoded result: jgnnq yqtnf

Do you want to run this program again?

>>Type 'yes' or 'no': yes

Type 'encode' to encrypt, type 'decode' to decrypt:

>>decode

Type your message:

>>jgnnq yqtnf

Type the shift number:

>>2

Here's the decoded result: hello world

Do you want to run this program again?

>>Type 'yes' or 'no': no

Goodbye.

Day 6: CAESAR HACKER

This program can hack messages encrypted with the Caesar cipher from Day 5, even if you don't know the key. There are only 26 possible keys for the Caesar cipher, so a computer can easily try all possible decryptions and display the results to the user. In cryptography, we call this technique a brute-force attack. If you'd like to learn more about ciphers and code breaking, you can read book *Cracking Codes with Python* (No Starch Press (2018; <https://nostarch.com/crackingcodes/>)).

Program in Action:

Caesar Cipher Hacker,

Enter the encrypted Caesar cipher message to hack.

>> QIIX QI FC XLI VSWI FYWLIW XSRMKLX.

Key #0: QIIX QI FC XLI VSWI FYWLIW XSRMKLX.

Key #1: PHHW PH EB WKH URVH EXVKHV WRQLJKW.

Key #2: OGGV OG DA VJG TQUG DWUJGU VQPKIJV.

Key #3: NFFU NF CZ UIF SPTF CVTIFT UPOJHIU.

Key #4: MEET ME BY THE ROSE BUSHES TONIGHT.

Key #5: LDDS LD AX SGD QNRD ATRGDR SNMHFGS.

Key #6: KCCR KC ZW RFC PMQC ZSQCFCQ RMLGEFR.

--snip--

Day 7: The Factor Finder

A number's factors are any two other numbers that, when multiplied with each other, produce the number. For example, $2 \times 13 = 26$, so 2 and 13 are factors of 26. Also, $1 \times 26 = 26$, so 1 and 26 are also factors of 26. Therefore, we say that 26 has four factors: 1, 2, 13, and 26. If a number only has two factors (1 and itself), we call that a prime number. Otherwise, we call it a composite number.

Program in Action:

Factor Finder,

--snip--

Enter a number to factor (or "QUIT" to quit):

>> 26

1, 2, 13, 26

Enter a number to factor (or "QUIT" to quit):

>> 4352784

*1, 2, 3, 4, 6, 8, 12, 16, 24, 29, 48, 53, 58, 59, 87, 106, 116, 118, 159,
174, 177, 212, 232, 236, 318, 348, 354, 424, 464, 472, 636, 696, 708, 848,
944, 1272, 1392, 1416, 1537, 1711, 2544, 2832, 3074, 3127, 3422, 4611, 5133,
6148, 6254, 6844, 9222, 9381, 10266, 12296, 12508, 13688, 18444, 18762, 20532,
24592, 25016, 27376, 36888, 37524, 41064, 50032, 73776, 75048, 82128, 90683,
150096, 181366, 272049, 362732, 544098, 725464, 1088196, 1450928, 2176392,
4352784*

Enter a number to factor (or "QUIT" to quit):

>> 9787

1, 9787

Enter a number to factor (or "QUIT" to quit):

>> quit

Day 8: The Bagels

In Bagels, a deductive logic game, you must guess a secret three-digit number based on clues. The game offers one of the following hints in response to your guess: “Pico” when your guess has a correct digit in the wrong place, “Fermi” when your guess has a correct digit in the correct place, and “Bagels” if your guess has no correct digits. You have 10 tries to guess the secret number.

Program in Action:

Bagels, a deductive logic game.

I am thinking of a 3-digit number. Try to guess what it is.

Here are some clues:

When I say:

Pico --> One digit is correct but in the wrong position.

Fermi --> One digit is correct and in the right position.

Bagels --> No digit is correct.

I have thought up a number.

You have 10 guesses to get it.

Guess #1:

>> 123

Pico

Guess #2:

>> 456

Bagels

--snip--

Fermi Fermi

Guess #8:

>> 701

You got it!

Do you want to play again? (yes or no)

>> no

Thanks for playing!

Day 9: PyPassword Generator

It generates password based on user's need. This asks user for no of letters, numbers and special characters and randomizes them to create a password. It also checks strength of password by counting total no of characters and categorizes them as weak (≤ 6), medium (7) and strong (≥ 8) and informs user. You can increase the uncertainty in positions of character to test your higher skill

Program in action:

Welcome to the PyPassword Generator!

How many letters would you like in your password?

>>6

How many symbols would you like?

>>2

How many numbers would you like?

>>1

*Here is your password: *+4oHJUJA*

Your password is strong.

Day 10: The Hacking Minigame

In this game, the player must hack a computer by guessing a seven-letter word used as the secret password. The computer's memory banks display the possible words, and the player is given hints as to how close each guess was. For example, if the secret password is MONITOR but the player guessed CONTAIN, they are given the hint that two out of seven letters were correct, because both MONITOR and CONTAIN have the letter O and N as their second and third letter. This game is similar to Project "Bagels" of Day 8.

Program in Action:

Hacking Minigame,

Find the password in the computer's memory:

```
0x1150 $J,>@|~~RESOLVE^      0x1250 {>+)<!?CHICKEN,%      0x1170 _;)][#?<&~$~+&}}
0x1160 }@%_-.:/$^|<|!(      0x1260 .][})?#@#ADDRESS      0x1270 ,#=>{-;/DESPITE
0x1180 %[!]{REFUGEE@?~,      0x1280 }/,}!-DISPLAY%%/      0x1190 _[^%[@}'^<_+{_@$~
```

```
0x1290 =>>,:*%?_?@+{)%#.      0x11a0 )?~/)+PENALTY?-=      0x12a0 >[,?*#IMPROVE@$/  
--snip--  
Enter password: (4 tries remaining)  
> resolve  
Access Denied (2/7 correct)  
Enter password: (3 tries remaining)  
> improve  
ACCESS  GRANTED
```

Day 11: Cho-han

Cho-han is a dice game played in gambling houses of feudal Japan. Two six-sided dice are rolled in a cup, and gamblers must guess if the sum is even (cho) or odd (han). The house takes a small cut of all winnings. The simple random number generation and basic math used to determine odd or even sums make this project especially suitable for beginners.

Program in Action:

Cho-Han

In this traditional Japanese dice game, two dice are rolled in a bamboo cup by the dealer sitting on the floor. The player must guess if the dice total to an even (cho) or odd (han) number.

You have 5000 mon. How much do you bet? (or QUIT)

>> 400

The dealer swirls the cup and you hear the rattle of dice.

The dealer slams the cup on the floor, still covering the dice and asks for your bet.

CHO (even) or HAN (odd)?

>> cho

The dealer lifts the cup to reveal:

GO - GO

5 - 5

You won! You take 800 mon.

The house collects a 40 mon fee.

--snip--

Day 12 : Higher Lower

Higher or lower is game based on making guess. User will be given name and description of two famous people or company and asked who has more followers. Program counts no of correct answer and game ends once user has given wrong answer.

Note: You will be provided with a list of dictionaries containing name, description and followers to write the program

Program in action:

Compare A: Ellen DeGeneres, a Comedian, from United States.

Compare B: Dwayne Johnson, a Actor and professional wrestler, from United States.

>>Who do you think has more followers? Type 'a', or 'b': b

Correct! Your current score is: 1.

Compare A: 9GAG, a social media platform, from China.

Compare B: Dwayne Johnson, an Actor and professional wrestler, from United States.

>>Who do you think has more followers? Type 'a', or 'b': b

Correct! Your current score is: 2.

Compare A: LeBron James, a Basketball player, from United States.

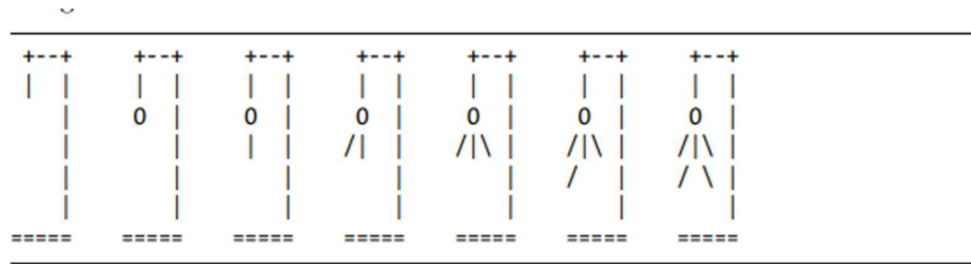
Compare B: Dwayne Johnson, a Actor and professional wrestler, from United States.

Who do you think has more followers? Type 'a', or 'b': a

Incorrect, your final score is 2.

Day 13: Hangman

This classic word game has the player guess the letters to a secret word. For each incorrect letter, another part of the hangman is drawn. Try to guess the complete word before the hangman completes. The secret words in this version are all animals like RABBIT and PIGEON, but you can replace these with your own set of words. The HANGMAN_PICS variable contains ASCII-art strings of each step of the hangman's noose:



Program in action:

When you run *hangman.py*, the output will look like this:

Hangman, by Al Sweigart al@inventwithpython.com

```
+--+
|  |
|  |
|  |
|  |
=====
The category is: Animals

Missed letters: No missed letters yet.

_ _ _ _ _
Guess a letter.
> e
--snip--
+--+
|  |
|  |
0  |
/  |
|  |
|  |
=====
The category is: Animals

Missed letters: A I S
O T T E _
Guess a letter.
> r
Yes! The secret word is: OTTER
You have won!
```

Day 14: Blackjack

Blackjack, also known as 21, is a card game where players try to get as close to 21 points as possible without going over. This program uses images drawn with text characters, called ASCII art. American Standard Code for Information Interchange (ASCII) is a mapping of text characters to numeric codes that computers used before Unicode replaced it. The playing cards in this program are an example of ASCII art:

```
|A| |10|
|♣| |♦|
|_A| |_10|
```

Program in Action:

When you run *blackjack.py*, the output will look like this:

```
Blackjack, by Al Sweigart al@inventwithpython.com

Rules:
    Try to get as close to 21 without going over.
    Kings, Queens, and Jacks are worth 10 points.
    Aces are worth 1 or 11 points.
    Cards 2 through 10 are worth their face value.
    (H)it to take another card.
    (S)tand to stop taking cards.
    On your first play, you can (D)ouble down to increase your bet
    but must hit exactly one more time before standing.
    In case of a tie, the bet is returned to the player.
    The dealer stops hitting at 17.

Money: 5000
How much do you bet? (1-5000, or QUIT)
> 400
Bet: 400

DEALER: ???

|##| |2|
|###| |♥|
|_##| |_2|

PLAYER: 17

|K| |7|
|♦| |♦|
|_K| |_7|

(H)it, (S)tand, (D)ouble down
> h
You drew a 4 of ♦.
--snip--
DEALER: 18

|K| |2| |6|
|♦| |♥| |♣|
|_K| |_2| |_6|

PLAYER: 21

|K| |7| |4|
|♦| |♦| |♦|
|_K| |_7| |_4|

You won $400!
--snip--
```
