**Problem 1 : Guessing Game**

🡪IPO Chart

|  |  |
| --- | --- |
| Input | Guess and input a number between 1 to 100 |
| Processing | If guessed number is incorrect then give another chance until correct answer.  Keep track of guesses of the user. |
| Output | Correct answer and number of guesses |

🡪Pseudocode

#GET input

#SET the initial values

#IF number == randomNumber

#DISPLAY break

#IF number < randomNumber

#DISPLAY output saying sorry, my number is higher than that and try again

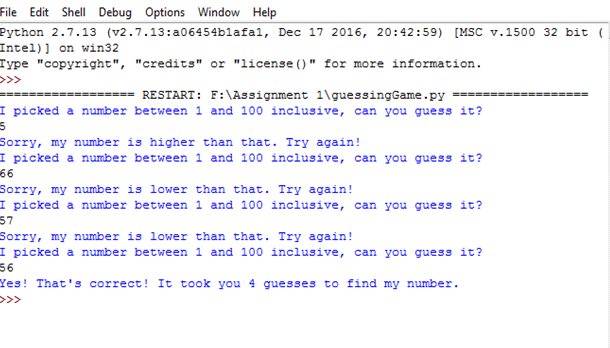
#IF number > randomNumber

#DISPLAY output saying sorry, my number is lower than that and try again

#ELSE

#DISPLAY output and give the guesses taken by the user

🡪Screenshot of the program



🡪Trace

|  |  |
| --- | --- |
|  | Test Data |
| #SET the initial values | 5 |
| #IF number == randomNumber | 56 |
| #DISPLAY break | - |
| #IF number < randomNumber | 66 |
| #IF number > randomNumber | 5 |
| #DISPLAY output and give the guesses taken by the user | 4 |

**Problem 2 : String Centering**

🡪IPO Chart

|  |  |
| --- | --- |
| Input | Input a short message but no longer than 30 characters |
| Processing | If message is longer than 30 characters then display that it is too long and end the program. |
| Output | If input is short then output on a 80 character line display the centered message |

🡪Pseudocode

#GET input

#CALCULATE string length

#IF input length > 30

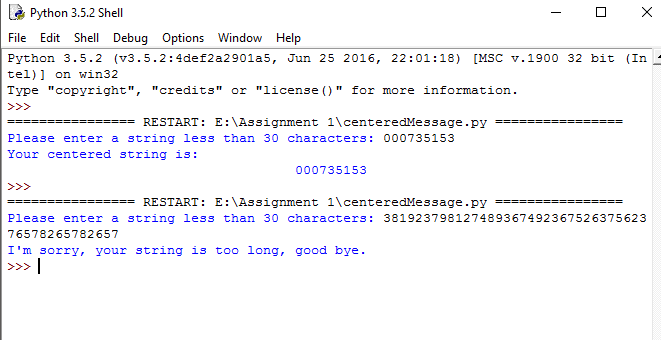
#DISPLAY message which tells string too long

#GET new input

#CALCULATE centerMessage

#DISPLAY output

🡪Screenshot of the program



🡪Trace

|  |  |
| --- | --- |
|  |  |
| #GET input | 3819237981491965626349979433498 |
| #CALCULATE string length | stringLength = len(userString) |
| #IF input length > 30 | if (stringLength < 30) : |
| #DISPLAY message which tells string too long | I'm sorry, your string is too long, good bye |
| #GET new input | 000735153 |
| #CALCULATE centerMessage | print ( " " \* (40 - centerMessage) + userString) |
| #DISPLAY output | 000735153 |

Problem 3 : Plinko!

🡪IPO Chart

|  |  |
| --- | --- |
| Input | Pick a random number from 1 to 5 inclusive |
| Processing | Drop a disk into a game board  Disk moves down and hits 5 pins  When disk hit the pins it generates a random number between 1 and 100.  If number <= 50 then ”tick”  If number >= 50 then ”tock”  After 5 pin hits the total number of left and right determines the prize to the winner.  Add this number to total prize winnings and repeat for each disk player |
| Output | Amount won at each disk drop and comments  Total prize to the winner |

🡪Pseudocode

#GET input

#GET the value of a and b

#WHILE a <= randomNumber

#PRINT the order of dropping disk

# WHILE b < 5

#IF number <= 50

#PRINT -tick-

#ELSE display -tock-

#PRINT totaSum

#IF totalSum == 0 and total = 10000

#PRINT output saying that disk landed in $10000 , congratulations

#ELIF total sum is in -1 , 1 , -5 , 5 and total = 0

#PRINT output saying that the disk is landed in $0

#ELIF total sum is in -2 , 2 and total = 1000

#PRINT output saying that the disk is landed in $ 1000 , congratulations

#ELIF total sum is in -3 ,3 and total = 500

#PRINT output saying that the disk is landed in $ 500 , congratulations

#ELIF total sum is in -4 , 4 and total = 100

#PRINT output saying that the disk is landed in $100 , congratulations

#CALCULATE newTotal

#DISPLAY output

🡪Screenshot of the program

