Mini Project's problem statements:

In every project you have to build a **TUI (Text based user interface)**

(No GUI and backed is to be used)

Project 1:

Design a project where as an input student will give a static number (between 1 to 6) and then roll the dice which randomly generate some value between 1 to 6. The winning situation arrives when the given static/fixed number exactly same to the number came after rolling the dice.

User can play the game as many numbers of times he wants until user wants to exit, and whenever winning situation occur some scores must be given to the user, and these scores goes on adding if user play this game multiple number of times.

Note: Dice contains face value's (1 to 6)

Hint: Make use of random.randint() function

(Student is free to decide the input and output layout for this mini project)

Project 2:

Create the Rock, Paper and Scissors game with Python, we need to take the user's choice and then we need to compare it with the computer choice which is taken using the random module in Python from a list of choices, and if the user wins, then the score will increase by 1.

Example:

User: Rock, Paper or Scissors?
CPU: rock
Tie
User: Rock, Paper or Scissors?
CPU: paper
You win! Paper covers Rock
User: Rock, Paper or Scissors?
CPU: scissors
You lose… Rock smashes Scissors
Rock, Paper or Scissors?
End the game
Final Scores:
CPU:1
User:1

Hint: Make use of random module to design the game

(Student is free to decide the input and output layout for this mini project)

Project 3:

Email Slicer is just a simple tool that will take multiple email address as an input and slice it to produce the username and the domain associated with it. The email must be divided into two strings by using '@' as the separator.

So, user provides n number of email addresses and you have to design a logic to slice the username and the domain out of those email addresses. The domain part must print in capitals.

Note: Email addresses must be stored first in some container and then operate the required logic on it.

Example:

abc@gmail.com

xyz@yahoo.com

after slicing

username :abc and domain: **GMAIL.COM**

username: xyz and domain: **YAHOO.COM**

(Student is free to decide the input and output layout for this mini project)

Project 4:

Create a program that takes the length of the password as input and generates a random password of the same length. The strength of the password depends equally on the 4 properties mentioned below. If the password generated randomly following the rules or constraints given below, then that password is treated as good in terms of strength and accepted otherwise ignore that password.

The properties to be followed for a strong password are:

- At least 12 characters.

- A mixture of both uppercase and lowercase letters.

- A mixture of letters and numbers.

- Inclusion of at least one special character, e.g., @ #?]

Note: do not use < or > in your password, as both can cause problems in Web browsers.

(Student is free to decide the input and output layout for this mini project)

Project 5:

The task is to create a script that generates a random number between a fixed range, and if the user guesses the number right in **three chances**, then user wins otherwise user lose.

This game user can play as many numbers of times and whenever user wins a score is to be given to the user.

At the end the users score must be displayed on the screen.
Hint: Make use of random module to design the game

Abstract steps:

- The user enters the lower and upper bounds of the range.
- As a result, the compiler generates a random integer between the range and stores it in a variable for future use.

- A while loop will be created for repetitive guessing.
- When a user guesses a number that is greater than a randomly selected number, the user receives the message "**have one more try**". Your guess was too high.
- If the user guesses a number smaller than a randomly selected number, the user gets an output of "**have one more try** ". Your guess was too small"
- In addition, if the user guesses within a minimum number of attempts, they get a "**Congrat's**" message and also get the winning scores.
- If the user fails to guess the integer in the minimum number of guesses, he/she will receive a "Better Luck Next Time!

(Student is free to decide the input and output layout for this mini project)

Project 6:

You need to write a python script that generates an acronym word from a given sentence.
- Take multiple strings as input in the form of list.
- Add the first letter of each string to output.
- Iterate over the complete string and add every next letter to space to output.
- Change the output to uppercase (required acronym).
- You have to generate acronyms for all given strings.

(Student is free to decide the input and output layout for this mini project)

Hint: You can use split and indexing to fetch the first word and then combine it.

Project 7:

The task is to generate a random story every time user runs the program.
Every time the user runs the program, we must produce a random tale. We'll first store the portions of the tales in distinct lists, and then use the Random module to choose random sections of the stories from those lists:

To construct a random narrative, we first imported the random module, **then built sections of the stories in separate lists**, then randomly picked portions of the lists.

Note: You store the portion of your tale/story in different lists, and during displaying the story when you pick the portions randomly your complete story must make some sense.

(Student is free to decide the input and output layout for this mini project)

Hint:

- Random module can be used to select random parts of the story stored in different lists.
- Sections of the story can be an adjective, a preposition, a proper noun, etc.

Project 8:

Your task is to build a currency converter that will allow you to convert currencies from one unit to another, such as converting Indian rupee into pounds, euros, US dollar, Canadian dollar, Chinese yuan, Russia's Rubal, etc. or vice versa.

In this project build a TUI (Text based user interface), where you will enter the source currency to be converted and conversion rate. And after conversion display the amount in the target currency.

(Student is free to decide the input and output layout for this mini project)

Project 9:

You have to build a dictionary (Or any other container of your choice) which contains multiple True/false type quiz questions.

Every participant/user will attempt 5 rounds and in each round random quiz questions will be displayed to the user/participant.

If the participant answers the quiz question correct, then congratulate him and add the scores.

At the end display the details and score of the participant.

(Student is free to decide the input and output layout for this mini project)

Project 10:

You task is to build a scientific calculator that performs all the below listed functionalities.

1. Add, sub, multiply, divide, and mod (%) operations on entered integer or floating type numbers.
2. Square root, exponent (power (a, b))
3. Sine, cosine, and tangent (Trigonometric functions).
4. Conversion from radian to degree and degree to radian.

   Above listed operations user can perform as many numbers of times until user hits the exit.

(Student is free to decide the input and output layout for this mini project)

Project 11:

In this project you have to enter a range [A, B] and system will randomly pick any number from your given range and check the status of that given number.

The properties to be checked are:

1. Is that number is odd or even
2. Is that number is positive or negative number
3. Is that number is prime number or composite number.

After checking the status, you have to display all the properties followed by the randomly picked number.

For example

Range is (1,12) and randomly picked number is 10

Then the properties followed by this number are:

10 is a positive number

10 is an even number

10 is a composite number

(Student is free to decide the input and output layout for this mini project)

Project 12:

In this project user will enter single or multiple numbers and your system will predict that the entered number or number's is/are valid number(s) in a Fibonacci series or not.

For example, if user inputs 2 numbers

0 and 7

0 is valid and 7 is invalid.

Because if we plot Fibonacci series 0 1 1 2 3 5 8 13……, In this series 0 is their but 7 is not present.

Constraint: range of the single number or multiple numbers you are entering can be huge.

(Student is free to decide the input and output layout for this mini project)

Project 13:

Create a multiplication table application where user will enter a sentinel value n and the application will display the mathematical multiplication tables till given sentinel value **n**.

For example, if user enters n = 4 then application will display the multiplication tables of 2, 3, and 4.

Constraint:

- Make use of oop concepts class methods and attributes

(Student is free to decide the input and output layout for this mini project)

Project 14:

You have to create a repository of your classmate's name and mobile number's. And whenever you want to search the mobile/contact number of some of your classmate, you can easily search it out from the repository you maintained.

Searching single contact is one of the functionalities, some other functionalities your app must cover are:

1. Your app must be capable of displaying all the contacts with your classmate names present in the repository.
2. Your app must be capable of extracting contact numbers of your multiple classmates when required.

(Student is free to decide the input and output layout for this mini project)

Project 15:

In this project you have to enter a positive integer range [A, B] and system will find out the status (Prime or composite) of each number available in the given range. At the end print the count also.

Note: Make use of efficient approach to check prime status of the number.

For example:

Range is (7,10)

Then the status of each number in the range is:

7 is prime

8 is composite or not prime

9 is composite

10 is composite

Count: 1 prime and 3 composite numbers in the range.

(Student is free to decide the input and output layout for this mini project)

Project 16:

Design an application where user will input two dates.

1. His/her date of birth in DD/MM/YY format.
2. Current (Present day) date in DD/MM/YY format.
   Your app will calculate and let the user know **how many days** that person survived in this world.

Note: Your calculation must be accurate and you have to consider leap and non-leap years separately.

(Student is free to decide the input and output layout for this mini project)

Project 17:   **(Converter to convert given Roman to integer)**

Editorial/Explanation about Roman numbers:

Roman numerals are represented by seven different symbols: I, V, X, L, C, D and M.

Symbol	Value
I	1
V	5
X	10
L	50
C	100
D	500
M	1000

For example, 2 is written as II in Roman numeral, just two ones added together. 12 is written as XII, which is simply X + II. The number 27 is written as XXVII, which is XX + V + II.

Roman numerals are usually written largest to smallest from left to right. However, the numeral for four is not IIII. Instead, the number four is written as IV. Because the one is before the five, we subtract it making four. The same principle applies to the number nine, which is written as IX. There are six instances where subtraction is used:

I can be placed before V (5) and X (10) to make 4 and 9.

X can be placed before L (50) and C (100) to make 40 and 90.

C can be placed before D (500) and M (1000) to make 400 and 900.

Your task is create a converter which converts given roman numeral to an integer.

 (Student is free to decide the input and output layout for this mini project)

Statement 18: (Converter to convert given integer to Roman)

Editorial/Explanation about Roman numbers:

Roman numerals are represented by seven different symbols: I, V, X, L, C, D and M.

Symbol   Value

I	1
V	5
X	10
L	50
C	100
D	500
M	1000

For example, 2 is written as II in Roman numeral, just two ones added together. 12 is written as XII, which is simply X + II. The number 27 is written as XXVII, which is XX + V + II.

Roman numerals are usually written largest to smallest from left to right. However, the numeral for four is not IIII. Instead, the number four is written as IV. Because the one is before the five, we subtract it making four. The same principle applies to the number nine, which is written as IX. There are six instances where subtraction is used:

I can be placed before V (5) and X (10) to make 4 and 9.

X can be placed before L (50) and C (100) to make 40 and 90.

C can be placed before D (500) and M (1000) to make 400 and 900.

Your task is create a converter which converts given roman numeral to an integer.

 (Student is free to decide the input and output layout for this mini project)

Project 18:

Clock Angle Problem

Clock Angle Problem: Given time in hh:mm format in 24-hour notation, calculate the shorter angle between the hour and minute hand in an analog clock.

Input: 5:30

Output: 15°

Input: 21:00

Output: 90°

Input: 12:00

Output: 0°

(Student is free to decide the input and output layout for this mini project)

Project 19:

Your task is to write a program to find the **nth prime palindrome number,** n is the input user will give.

A prime number, such as 127, has no factors other than itself and one. A palindrome, such as 121, is the same number when its digits are reversed. A prime palindrome, such as 131, is both prime and a palindrome.

(Student is free to decide the input and output layout for this mini project)

Project 20:

A basket is given to you in the shape of a matrix. If the size of the matrix is N x N then the range of number of eggs you can put in each slot of the basket is 1 to N^2 . You task is to arrange the eggs in the basket such that the sum of each row, column and the diagonal of the matrix remain same.

Test case for your reference:

Input by the user dimension of the basket i.e., N = 3

So, number of eggs you can put at each slot are in the range of 1 to 3^2 (1 to 9)

Input:
6 3 6
5 5 5
4 7 4
Explanation:
   Now the value of the sum of
   any row or column as well as diagonal is 15
Note: 2 < = N <= 100

(Student is free to decide the input and output layout for this mini project)

Project 21:

Your task is to find if it is possible to cut the cake in the below mentioned ways for a given value of N.

Given an integer N and a cake which can be cut into pieces, each cut should be a straight line going from the center of the cake to its border. Also, the angle between any two cuts must be a positive integer. Two pieces are equal if their appropriate angles are equal.

The given cake can be cut in following three ways:

- Cut the cake into N equal pieces.
- Cut the cake into N pieces of any size.
- Cut the cake into N pieces such that no two of them are equal.

(Student is free to decide the input and output layout for this mini project)

Project 22:

Your task is to find the name of the student with maximum marks after updation in marks and the jump in the student's rank i.e., previous rank – current rank.

You are given three **lists' names, mark's and update's** where:

- Names contain the names of students.
- Marks contain the marks of the same students.
- Updates contains the integers by which the marks of these students are to be updated.

   (Number of levels a student is ranking up or down must be displayed)

(Student is free to decide the input and output layout for this mini project)

Project 23:

You task is to replicate the working of ATM for a single user, let's say, Mr. John, who has already successfully logged into her account on the ATM Machine, now, we will program the next steps he may want to perform.

Like display his name and cash available in his savings account

Withdraw cash and display the status of the cash balance.

Deposit cash and display the status of the cash balance.

(Your task is to design and implement the ATM functionality by taking care of all constraints, for example if minimum cash available is less than 5000 then system will display low balance)

(Student is free to decide the input and output layout for this mini project)

Project 24:

Your task to create a functionality in which when user will input a range of two dates. Then your module will find and print all years in the range of given dates those are leap years separately and rest of the years those are non-leap separately.

For example:

Input date range in the format dd/mm/yy

(12/01/200) to (13/12/2025)

Leaps years are:

2000, 2004, 2008, 2012, 2016, 2020, 2024, 2028, 2032, 2036, 2040, 2044, 2048.

Non leap years are:

2001,2002,2005,2006,2007--

(Student is free to decide the input and output layout for this mini project)