Smart Programmers CHALLENGE

2024

Smart Programmers Python Challenge

RULE BOOK

GLOBAL ROBOTICS CHALLENGE

Smart programmers (python)

Contents

1. Python challenges

Considering programming's growing importance as a global skill, we are pleased to introduce the Smart Programmers competition for individuals aged 8 and above. The Smart Programmers competition consists of multiple rounds in each category, each featuring distinct challenges. It is mandatory for every team/member to complete the challenge and submit their work within a specified time frame.

1.1 Coder age

Junior Level (10 - 14 years)

Senior Level (15 – 19 years)

1.2 Team members

Team's members should be at maximum 3 members guided by a Coach/Teacher

It is not possible for teams to share a team member.

2. Rules

2.1 General rules

1) Every member is required to bring their own laptop.
2) The code/task must be submitted prior to the countdown.
3) During the competition time, accessing the internet is prohibited.
4) Teams will present their work to the judges once the round time is over.
5) The code created by the team for each challenge will be manually reviewed by the judges after the completion of each mission. It will be evaluated based on the judging criteria.
6) There will be a 15-minute period before each round to explain the challenge and allow team members to ask questions about it.

7) Any form of communication between team members and non-team members is strictly prohibited during the competition.
8) Interference or communication by mentors during the competition will result in a warning for the first offense, and repeated instances may lead to the team being potentially eliminated from the competition.

2.2 Python programmer's rules:

1) Before proceeding with these rules, please ensure you have read the general rules as they serve as the foundation for all other rules.
2) The acceptable age range is from 10 years old, divided into two main categories: junior (10-14 years) and seniors (15-18 years).
3) **Junior** Teams should use python without built-in function.
4) Senior Teams allowed using any built-in functions.
5) Challenges will be varied like implementing a specific algorithm, solving a mathematical problem, writing a code to address a given scenario.
6) Internet is not allowed.
7) Three rounds for each category time for each round **45 min**.

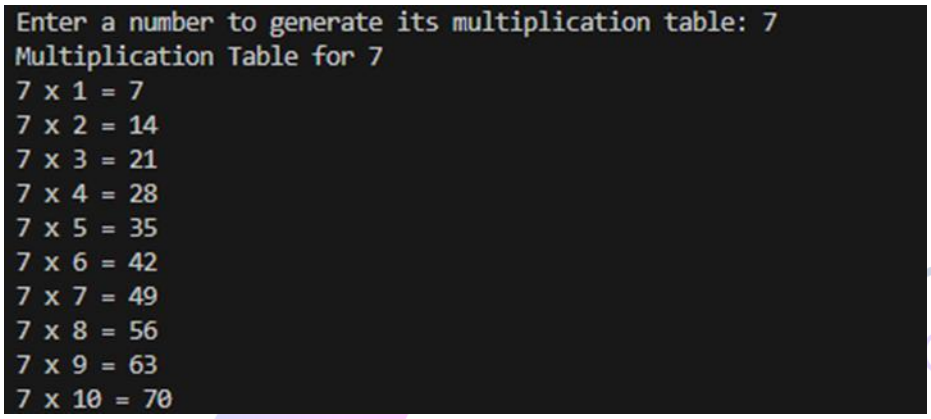
3. Samples

3.1 Samples of junior

Mission1: Multiplication Table Generator

You are a programmer and want to create a program that generates a multiplication table for a given number. The program should ask the user for a number, and then it should generate and display the multiplication table for that number from 1 to 10.

Expected output:

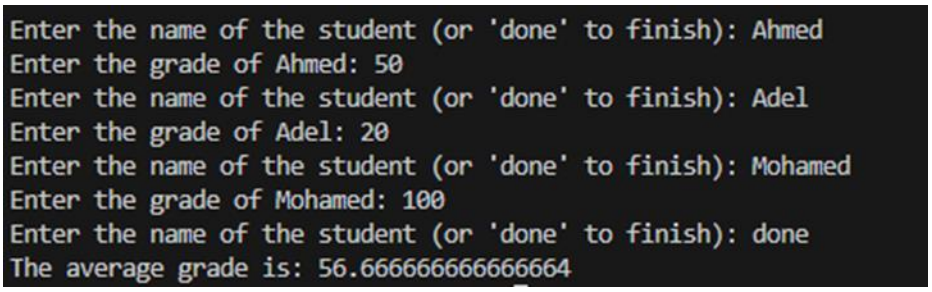


```
Enter a number to generate its multiplication table: 7
Multiplication Table for 7
7 x 1 = 7
7 x 2 = 14
7 x 3 = 21
7 x 4 = 28
7 x 5 = 35
7 x 6 = 42
7 x 7 = 49
7 x 8 = 56
7 x 9 = 63
7 x 10 = 70
```

Mission 2: Calculating Grade Average

You are a teacher, and you need to calculate the average grades of your students. You have a list of student names and their respective grades. Your task is to write a Python program that calculates and displays the average grade for the class.

Expected output:



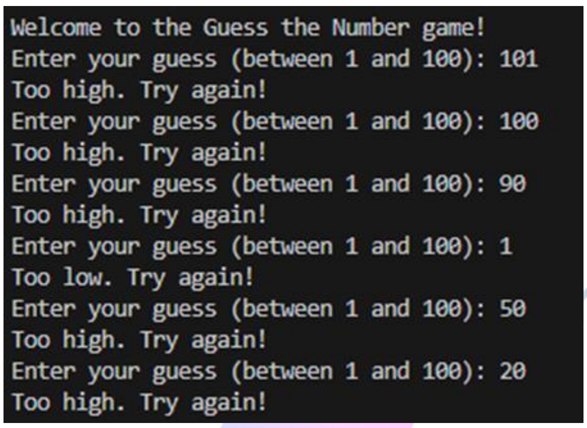
```
Enter the name of the student (or 'done' to finish): Ahmed
Enter the grade of Ahmed: 50
Enter the name of the student (or 'done' to finish): Adel
Enter the grade of Adel: 20
Enter the name of the student (or 'done' to finish): Mohamed
Enter the grade of Mohamed: 100
Enter the name of the student (or 'done' to finish): done
The average grade is: 56.666666666666664
```

3.2 Samples For Senior level:

Mission 1: Guess the Number Game

You are a programmer and wants to create a simple "Guess the Number" game where the program will generate a random number between 1 and 100, and the player must guess the number. The program will provide feedback to the player, indicating whether their guess is too high, too low, or correct. The game will continue until the player guesses the correct number.

Expected output:

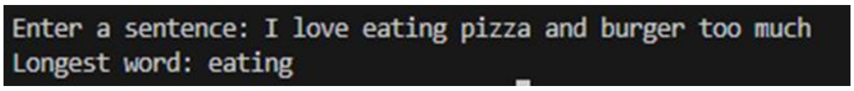


```
Welcome to the Guess the Number game!
Enter your guess (between 1 and 100): 101
Too high. Try again!
Enter your guess (between 1 and 100): 100
Too high. Try again!
Enter your guess (between 1 and 100): 90
Too high. Try again!
Enter your guess (between 1 and 100): 1
Too low. Try again!
Enter your guess (between 1 and 100): 50
Too high. Try again!
Enter your guess (between 1 and 100): 20
Too high. Try again!
```

Mission 2: Finding the Longest Word in a Sentence

You are a programmer and tasked with writing a Python program that finds the longest word in each sentence. The program should take a sentence as input, analyze the words within the sentence, and output the longest word.

Expected output:



```
Enter a sentence: I love eating pizza and burger too much
Longest word: eating
```