**Derive an algorithm for efficiently predicting the Coding skill score of a student in a particular Language given the following data.**

**Input Format** **(File):**

* First line of the input file is name of the Student.
* Second line of the input is n number of tests attended by a student.
* Following n\*2 number of lines
* first line consists of space separated inputs of:
  + Total score of the question.
  + Programming language (JAVA, C, C\_PLUS\_PLUS, DOT\_NET, PYTHON).
  + Question Type (PROGRAM, MCQ, FILL\_UP).
  + Difficulty Level (LOW, MEDIUM, HIGH).
  + Average time taken.
  + No. of correct responses.
  + NO. of wrong responses.
* second line consist of space separated inputs of:
  + Scored gained by a student.
  + Time taken by a student to answer the question.
  + No. of times programming question compiled.
  + No. of times MCQ answers were changed.
  + Answer of a student is correct or wrong (true/false).
* Next line consists of N number of learning videos viewed by a student
* following N lines of space separated:
  + Language of the video (JAVA, C, C\_PLUS\_PLUS, DOT\_NET, PYTHON).
  + No. of videos viewed.

**Algorithm Explanation:**

The score of the student is splitted on a basis of the type of the questions

|  |  |  |
| --- | --- | --- |
| **Type of question** | **Score for Efficient Answer** | **Score for Time Taken** |
| **MCQ** | 95% | 5% |
| **Fill-up** | 95% | 5% |
| **Program** | 80% | 20% |

* The time taken to complete an MCQ and fill-up does not play a major role on predicting the coding skill of the student, Because MCQ has choices we can choose any one of them and also fill-up doesn’t need a high skill to solve it.
* The Program questions are given more importance than an MCQ and Fill-up on the score calculation, Because the student should have high knowledge to answer the program questions than the MCQ and fill-up.

**Score Calculation for efficient answer:**

**MCQ:**

* If the total number of students who have attempted this question wrongly is greater than the number of students who have attempted this question correctly then the

Score =Score+ (number of students who have attempted this question wrongly/Total no of students attempted the question)

* Number of times answer was changed
  + For every 2 change in the answer the score reduces by 1.
* Difficulty level of the question:
  + Low – 1 percent hike on the score.
  + Medium – 2 percent greater than low
  + High – 3 percent greater than low.

**Fill-up:**

* If the total number of students who have attempted this question wrongly is greater than the number of students who have attempted this question correctly then the

Score =Score+ (number of students who have attempted this question wrongly/Total no of students attempted the question)

* Difficulty level of the question:
  + Low – 1 percent hike on the score.
  + Medium – 2 percent greater than low
  + High – 3 percent greater than low.

**Program:**

* If the total number of students who have attempted this question wrongly is greater than the number of students who have attempted this question correctly then the (Wrong answer is greater than 50%)

Score =Score+ (number of students who have attempted this question wrongly/Total no of students attempted the question)

* Number of times program was compiled
  + After first 10 compilation for every 10 compilation of the program the score is reduced by 1 percent.
* Difficulty level of the question:
  + Low – 1 percent hike on the score.
  + Medium – 2 percent greater than low
  + High – 3 percent greater than low.

**Score Calculation for time taken:**

* If the question answered by the student is correct then only the score for time taken will be awarded to the student.
* Based on the following scenario the scores are awarded to the students:
  + When the time taken by the student is greater than the average time taken by other students to solve this question:
    - When greater than the half of the average time then half score of time taken will be reduced from the total score.
    - When greater than 2 times of the average time then full score of time taken will be reduced from the total score.
  + When the time taken by the student is lesser than the average time taken by other students to solve this question:
    - when lesser than the half of the time the full score will be awarded to the student.
    - When greater than half of the time then half of score will be awarded to the student.

**Videos viewed:**

* If a student view more than 10 learning video the score will reduce by 1 for each 10 videos up to 5.

**Final Calculation:**

* Out of 100 the weightages of the questions are splitted as the follows:
  + Program- 85
  + Fill-up- 10
  + MCQ- 5

**Performance run metrics:**

* 100 question 100 attempts = 17 milliseconds.
* 10000 question 10000 attempts = 128 milliseconds.
* 100000 question for 1 million attempts = 619 milliseconds.

**Output:**

* The coding skill score of a student in a particular Language is displayed out of 100.

**Sample Input:**

//file//

Sabareesh

5

100 C\_PLUS\_PLUS MCQ MEDIUM 689 26 74

72 27 22 6 true

100 C FILL\_UP HIGH 802 7 93

38 3873 48 21 false

100 JAVA FILL\_UP HIGH 784 43 57

19 1394 39 5 false

100 JAVA MCQ LOW 890 71 29

43 161 14 25 true

100 C\_PLUS\_PLUS PROGRAM MEDIUM 1516 82 18

12 2374 13 21 false

4

JAVA 44

C 1

C\_PLUS\_PLUS 17

DOT\_NET 2

**sample output:**

Score of Sabareesh for

JAVA is : 20

C\_PLUS\_PLUS is : 29

C is : 38

The Program is Executed for : 1359 milliseconds