Проекти

ниво РНР 1, 2020/2021

Изисквания към проектите

1. Приложение разработено от един курсист с РНР.
2. Приложението трябва да изпълнява функционалностите описани в задачата.
3. Приложението трябва да бъде стилизирано.
4. Работата по проекта трябва да бъде отразена в GitHub - синхронизирането трябва да бъде разпределено равномерно в рамките на **поне една седмица** преди крайния срок за предаване на проектите.
5. В таблицата срещу вашето име, в колоната след номера на проекта ви, поставете линк към проекта си в GitHub
6. Краен срок за предаване на проектите - **30.01.2021, 23:59 ч.**

#1 IsSiStebinStack

[Si Stebbins](https://en.wikipedia.org/wiki/Si_Stebbins_stack)  е начин да подредите тесте с карти за игра -

* цветовете на картите следват тази последователност - [спатия, купа, пика, каро] или **[Clubs, Hearts, Spades, Diamonds]**
* всяка следваща карта е със стойност 3 по-голяма от предшестващата

За картите с нечислена стойност

* Асо/Aces/ - има стойност 1
* Вале/Jacks/ - 11
* Дама/Queens/ - 12
* Поп/Kings/ - 13

При подредба [Si Stebbins](https://en.wikipedia.org/wiki/Si_Stebbins_stack) щом се достигне стойност по-голяма от 13, броенето започва пак от 1.

**Например**.

В тесте, подредено по Si Stebbins - имаме

10 каро

Поп спатия

3 купа и т. н.

Като имате тесте от 52 карти /или по-малко/, вашата задача е да определите дали картите са подредени по [Si Stebbins](https://en.wikipedia.org/wiki/Si_Stebbins_stack) или не.

Тестето от карти е представено като стринг, в който отделните карти са разделени с интервал.

Например, стрингът "10D KC 3H" представлява тестето: 10 каро, Поп спатия, 3 купа (което, както видяхме е подредено по Si Stebbins).

**Като резултат** отпечатайте **1**/или друг подходящ отговор/, ако тестето е подредено по Si Stebbins.

Ако тестето не е подредено по Si Stebbins, **отпечатайте число от 2 до 51**, представляващо поредността на картата в тестето, при която подредбата по Si Stebbins е прекъсната.

*Забележка: В тестето с карти винаги имаме само валидни карти. Не е необходимо да проверявате за дублиращи се или невъзможни карти.*

[Повече информация и тестове](https://docs.google.com/document/d/1b3cblqW1dGOMJpmgqFCERBpcw7KAux7RPMDeJ8PKtsM/edit?usp=sharing)

#2 TrafficLights1D

Пътувате по дълъг прав път с много светофари. Пътували сте по този път много пъти и сте много наблюдателни. Вече знаете позициите /в метри/ и продължителност /в секунди/ на всеки светофар. Запазили сте данните в масива **roadMap**.

Пример, как са записани данните -

* [15, 9] представлява светофар, който се намира на 15 метра от началната точка на пътя и продължителността му е 9 секунди.

С други думи - светофарът свети в зелено 9 секунди, в червено - 9 секунди, после пак в зелено 9 секунди и т. н.

Получавате като **входни данни** масива **roadMap**. Вашата задача е да определите колко време ще ви е необходимо да стигнете края на пътя като се движите със скорост **1 метър** в секунда.

**Забележки:**

* Винаги започвате от позиция 0, във време 0.
* Таймера спира в момента, в който преминете последния светофар в списъка.
* В момент 0, всички светофари се превключват на зелена светлина.
* На пътя няма никой - освен светофарите нямате други препятствия, които да пречат на свободното ви придвижване по пътя.
* Пътувате с постоянна скорост 1 метър в секунда, така че не губите време за спиране или ускоряване; в момента, в който светне зелената светлина - тръгвате.

[Повече информация и тестове](https://docs.google.com/document/d/1MzdKP91ymk1SImi_qPnz3LAAEDQQUAQ5SCa3mmC9r5Y/edit?usp=sharing)

#3 Dresses

Диана планира да отиде на парти този уикенд! Иска да се изфука с нов вид, така че решава да си купи нова рокля. Тя е много капризна по отношение на цвета на бъдещата си рокля и наистина иска такава, която е повече зелена, отколкото синя или червена.

В магазина има няколко рокли и ние имаме **RGB** кода на техния цвят. Вашата задача е да помогнете на Диана да намери тези рокли, в които има повече зелено, отколкото синьо или червено - там където зеления цвят има най-висока стойност.

**Забележка:**

* **RGB** представянето на цвета е в 6-цифров хекс код. Първите две цифри се отнасят за червения компонент, 3 и 4 - за зеления компонент, последните са за синия компонент.
* Има повече зелено в цвета в сравнение с червено и синьо, само и единствено ако зеления компонент е повече, както от червения, така и от синия. Например **FC7A07** е подходящ цвят за Diana, но **90CACA** не е, тъй като зеления и синия компонент са с равни стойности.

[Тестове и повече информация](https://docs.google.com/document/d/1eqAlZ_qiK0YlZm64J5W4o8mQxlmk0l5m7Wc4sbrYFzM/edit?usp=sharing)

#4 ClockHandAngle2

Втренчили сте се в часовника, мислейки за задачата [clockHandAngle](https://docs.google.com/document/d/1TYFQWCoWn8-mXjvfnXyqh-tnvUfLv-N86Z-tNYpm5wQ/edit?usp=sharing), която наскоро решихте успешно. Чудите се след колко време, стрелките на часовника ще застанат, образувайки ъгъл със същата големина като сега и искате да напишете алгоритъм, който ще определи този момент.

Като имате ъгъл /в градуси/ и времето сега - timeNow, дадено във формат hh:mm:ss, вашата задача е да намерите кога стрелките на часовника /показващи час и минути/ ще застанат в същия ъгъл.

**Като резултат** дайте времето в същия формат, закръглено надолу към най-близката секунда.

[Тестове и повече информация](https://docs.google.com/document/d/1BfUUirgu5GEl6KgkmuDreOTaxPljQwXU6PtgFrDk7f0/edit?usp=sharing)

#5 StringChainReplacements

Получавате масив от стрингове и вашата задача е да намерите най-малкия брой промени в знаците, необходими за всеки първи знак на стринг да бъде еднакъв с последния знак от предхождащия го стринг.

Нямате право да променяте последователността на стринговете!

Пример

* За stringArray = ["abc", "def", "ghi"], резултатът трябва да бъде stringChainReplacements(stringArray) = 2.
* Трябва да направим две промени - да сменим d във втория елемент заради с в първия, и g в третия заради f във втория.

[Тестове и повече информация](https://docs.google.com/document/d/1YwaOJ0KdQowg2-POMkll2W1hZWre9w8bI3wesSZjGPY/edit?usp=sharing)

#6 MasterVsFighters

A local fight club has hired you to provide statistics for their fight championships.

The Championships always follow **the same rules**:

* A Master will fight against some other fighters with different characteristics.
* The Master always lets the fighter attack first.
* The Master attacks right after the fighter (if the Master still has some health points).
* The fighters come to battle one by one, from the first to the last.
* It's a 1 vs 1 fight, and if the first fighter loses, then the next one comes in, until the Master either loses or beats all the fighters.
* The Master is declared winner when all fighters have no health remaining.
* The Fighters are declared winners when the Masters health is grounded to zero.
* The Master and the Fighters have 3 attributes, they are:
  + Health (from 1 to 20)
  + Attack (from 6 to 10)
  + Resistance (from 1 to 5)
* During the fight, each attack deals damage equal to the attacker's attack stat minus the defender's resistance stat. *Note: since the attack stat is always greater than any resistance stat, there won't be a situation where the attack will deal zero or a negative amount of damage*.
* There are 3 types of fighters: Apprentice, Brute and Guardian.

**Your task** is to return the following information in the form of an array of strings:

* [0] = winner: string("Master" or "Fighters")
* [1] = fightersDefeated: string(amount of fighters that lost the fight)
* [2] = fightersRemain: string(amount of fighters that remains if master is defeated)
* [3] = masterHealth: string(amount of master's health at the end; can't be below 0)
* [4] = hitsByFighters: string(amount of times that fighters hit the master)
* [5] = hitsByMaster: string(amount of times the master hit the fighters)
* [6] = avgDamageDealtByFighters: string(average damage per hit dealt by fighters to master, rounded to 2 decimal places)
* [7] = avgDamageDealtByMaster: string(average damage per hit dealt by master to fighters, rounded to 2 decimal places)

Each type of fighter has a pre-defined set of attributes:

Fighters = {  
 apprentice: {  
 health: 5,  
 attack: 6,  
 resistance: 2   
 },  
 brute: {  
 health: 6,  
 attack: 8,  
 resistance: 2  
 },  
 guardian: {  
 health: 8,  
 attack: 6,  
 resistance: 5  
 }  
}

Given the list of fighters and the master's stats, your task is to return the statistical information described above.

Example

* For fighters = ["a","a","b"] and master =[14, 6, 4] the result should be masterVsFighters(fighters, master) = ["Fighters", "2", "1", "0", "6", "5", "2.67", "4.00"].
* The fight consists of 2 Apprentices and a Brute against a Master with health = 14, attack = 6 and resistance = 4.
* The Fighters attacks first, so the apprentice does his first hit. The Master is able to win the first battle against the apprentice and the second against the other apprentice, but when fights against the brute, his life is just 6 (because of previous battles), so the brute attacks with 8, the Master resists only 4, so the health now is 2. The master attacks, but the brute still stands (with 2 HP) and attacks again, the Master's health is now under 0, so he lost.
  + The winner is the "Fighters"
  + 2 fighters were defeated
  + 1 fighter remains
  + The master has 0 health points
  + The fighters landed 6 hits (2 hits for each fighter)
  + The master landed 5 hits before being defeated (2 hits to each apprentice, and 1 hit to the brute)
  + Since there were 4 apprentice hits (dealing damage of 6 - 4 = 2 each time) and 2 brute hits (dealing damage of 8 - 4 = 4 each time), the average is (2 + 2 + 2 + 2 + 4 + 4) / 6 = 2.67
  + Both apprentices and the brute have a resistance of 2, so all hits by the master dealt 6 - 2 = 4 damage, so the average damage is 4.00

[Тестове и повече информация](https://docs.google.com/document/d/1RDG09nJkknPWs0Bq-3SFHy5z1LNvTwScpAvIRGAymAg/edit?usp=sharing)

#7 countAPI

Имате списък с API заявки във формат */project/subproject/method*.

**Задачата ви** е да изчислите и отпечатате броя на заявките към всяка част от адреса /API endpoint/ под формата на дърво.

В това дърво *projects*, *subprojects*, and *methods* трябва да бъдат сортиране в реда, в който за подадени като входни данни. Резултатът трябва да се състои от няколко стринга. Всеки подпроект се подрежда под неговия родителски проект. Всеки метод се подрежда под подпроекта, към който е включен.

Стрингът, представляващ проект започва с --, стрингът представляващ подпроект започва с ----, методите започват с ------.

След отпечатването на проект, подпроект или метод поставете броя на заявките към този модул в кръгли скоби.

[Тестове и повече информация](https://docs.google.com/document/d/1nwwmld2vHBhxsVmVUCLgLey3o_pa0e0GmaOgOxoBMYk/edit?usp=sharing)

#8 questionCorrectionBot

Когато потребител зададе въпрос в Quora, системата автоматично се опитва да **подобри форматирането** на въпроса, преди да го покаже на потребителите.

Имате стринг като **входни данни** и **вашата задача** е да го форматирате, като се съобразявате със следните правила:

* Премахнете излишните разстояния
  + Премахнете разстоянията в началото и края на стринга, дублираните разстояния между думите
* Коригирайте разстоянията около пунктуацията
  + Всяка запетая има разстояние след нея, но не и разстояние преди нея.
* Въпросите започват с главна буква.
* Въпросите завършват с точно един въпросителен знак - не 0 или повече от един въпросителен знак.

[тестове и повече информация](https://docs.google.com/document/d/1_RQZRuQtbX6Pk8Q5FMmQc1IKbcjHhjfhwu-FXx6OXgE/edit?usp=sharing)