PHP Opdrachten

- Bankrekening controle *
- Password Hash - zoeken in groot text bestand ***
- Corona Simulatie in PHP **
- Lotto Spel *
- Patiëntgegevens in JSON File **
- Quizzes

Bankrekening controle *

Image result for iban nummer

Een bankrekeningnummer in Nederland bestaat niet uit zomaar willekeurige nummers. Niet elk nummer van 9 cijfers is een geldig en goed bankrekeningnummer. Een bankrekeningnummer bestaat een aantal velden, bijvoorbeeld:

NL 69 INGB **0123456789**

De eerste twee tekens NL, staan voor het land, 69 is een controlenummer en INGB is de bank (in dit geval ING). Dan volgt er een een rekeningnummer van 10 cijfers. Deze opdracht gaat voer deze laatste 10 cijfers (in donkerrood weergegeven).

Om typefouten te voorkomen, wordt er in Nederland de elfproef gedaan met rekeningnummers. Dit betekent dat de nummers in een rekeningnummer vermenigvuldigd worden met 10 terug naar 1 en als de som van die getallen gedeeld door 11 een geheel getal is, dan is het een geldig rekeningnummer.
Bijvoorbeeld:

Rekeningnummer: 012 34 56 789
Som: 0*10 + 1*9 + 2*8 + 3*7 + 4*6 + 5*5 + 6*4 + 7*3 + 8*2 + 9*1 = 165

(Omdat het cijfer maar 9 posities lang is wordt er een voorloop nul aan toegevoegd)

165/11 is een geheel getal want 11 x 15 = 165 en 165 % 11 = 0, omdat de rest van de deling 165/11, 0 is.

12 34 56 789 is dus een geldig rekeningnummer.

Opdracht

Maak een functie die een rekeningnummer meekrijgt (alleen maximaal 10 cijfers) en die true of false terug geeft. True als het een goed bankrekeningnummer is en false als het een niet goed bankrekeningnummer is.

Let op netjes inspringen en lever de code in een .php file aan in Teams.

Optioneel (extra punten)

1. Maak een form waarmee je een rekeningnummer kunt invullen, zodat het kan worden

gecontroleerd.

2. Bereid de code uit, zodat je controleer of er een volledig IBAN nummer of alleen het rekeningnummer wordt ingevoerd. Als er een IBAN nummer wordt ingevoerd, controleer deze dan ook door hier het rekeningnummer zelf uit te halen.

3. Maak het formulier wat mooier door CSS te gebruiken. Bootstrap kun je ook gebruiken. Misschien kun je bij het vorm ook uitleggen waarvoor het dient.

4. Maak deze opdracht van jouw portfolio.

--

Password Hash - zoeken in groot text bestand ***

Snel zoeken in password hash lijst. Je leert wat een zoek algoritme is en gaat een zoekalgoritme implementeren.

Wat ga je leren?

- met hash functies werken
- PHP en files lezen en doorzoeken
- een programmeer-opdracht opdelen in stapjes; een plan maken
- wat een zoeklagoritme is
- implementeren van een zoekalgortime

Inleiding

Er is een zeer lange lijst met password hashes. Je kan daarin zoeken of jouw password is gehacked. Dit kan je online doen, maar je wilt je super-goede-geheime password waarschijnlijk niet zomaar naar een of andere site sturen. We gaan dus onze eigen lokale password validation maken. En de uitdaging is wie de snelste code kan maken. Hoe snel kan jij een hash vinden in een half miljard password hashes. Ik kan je vertellen dat dat op een i5 laptop met SSD onder de paar seconden moet kunnen.

Have I been powned?

Kijk op https://haveibeenpwned.com/Passwords je kunt daar je password invullen en controleren of het een 'bekend' (dus gehacked) wachtwoord is. Deze wachtwoorden zijn onder hackers bekend en zijn dus eigenlijk niet (meer) veilig. Maar als je de controle wilt uitvoeren zonder jouw wachtwoord over het internet te sturen dan zal je zelf iets moeten maken. En ja, de site heeft een secure connectie via SSL, maar wat als de programmeur stiekem een lijstje bijhoud van alle wachtwoorden die worden geprobeerd?

Case

We gaan dus onze eigen versie maken van 'Have I been powned'. Daarvoor moet je het grote bestand met hashes downloaden. Deze staat op de genoemde site. Kies de versie in het format SHA-1 ordered by hash.

De file ziet er als volgt uit:

```
000000005AD76BD555C1D6D771DE417A4B87E4B4:4
00000000A8DAE4228F821FB418F59826079BF368:3
00000000DD7F2A1C68A35673713783CA390C9E93:630
00000001E225B908BAC31C56DB04D892E47536E0:5
00000006BAB7FC3113AA73DE3589630FC08218E7:2
00000008CD1806EB7B9B46A8F87690B2AC16F617:4
0000000A0E3B9F25FF41DE4B5AC238C2D545C7A8:15
0000000A1D4B746FAA3FD526FF6D5BC8052FDB38:16
0000000CAEF405439D57847A8657218C618160B2:15
0000000FC1C08E6454BED24F463EA2129E254D43:40
```

Elke regel bestaat uit een password hash, een dubbele punt en een getal. Dit laatste getal geeft aan bij hoeveel hacks het password is gevonden. Het enige dat je moet is dus een password hash zoeken.

Hints

Onder Linux kun je met het wc (word count) commando vrij snel bepalen hoeveel regels het bestand heeft. Je zult er snel achter komen dat gewoon de file doorlopen en regel voor regel kijken of je de juiste hash hebt gevonden niet werkt. Dit ligt natuurlijk wel aan de snelheid van jouw computer/laptop. Probeer maar eens in te schatten hoelang het duurt voordat je bijvoorbeeld 10% van de file heb doorzocht. Ik heb een algoritme genaakt dat op elke eenvoudige laptop de hash binnen 1 seconden vind.

Als je gaat nadenken over een strategie bedenk dan hoe jij het handmatig zou aanpakken?

Deel nu het probleem op in stapjes en test elke stapje. Wat zijn je stapjes en hoe wil je het aanpakken.

Maak het probleem eenvoudiger.

Als je er nog niet uitkomt, maak het probleem dan eenvoudiger. Bijvoorbeeld je hebt 20 hashes:

```
1200
1201
2011
2045
```

```
2234
3400
4000
4001
4010
4098
4099
5332
8020
8100
8201
8245
8376
8898
8999
9010
```

Stel je zoekt de hash van het een wachtwoord en de hash is 9500. Ga jij nu 20x maal kijken of de hash gelijk is of zie je dit sneller? En stel je zoekt naar de hash 8020, ga je dan alle hashes vergelijken of die je dit op een snellere manier. Probeer een plan te maken hoe je zelf in zo min mogelijk stappen bepaald of een hash in de lijst staat. Deze strategie kun je dan mogelijk ook toepassen op het grote bestand.

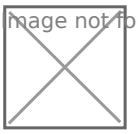
Maar een plan en bespreek dat eventueel met je docent voordat je begint. Een goed plan is het halve werk!

--

Corona Simulatie in PHP **

*In deze les gaan we code van een ander lezen en aanpassen. In het kader van Covid-19, (Corona) heb ik een virus-simulatie gemaakt. Dit is een **zeer vereenvoudigde weergave van de werkelijkheid** en je moet dit niet zien als een simulatie van de echte wereld. Dus alle getallen die je zien moet je niet betrekken op hetgeen er nu in de echte wereld afspeelt.*

De simulatie ziet er in de browser als volgt uit.



In het 'grid' (het grote gedeelte bovenin) zie je de bevolking. Elke karakter is een persoon. Een - is een gewoon, onbesmet persoon, een oranje s is een ziek persoon, een groene o is een immuun persoon en een rode X is een overleden persoon. Met de knoppen kun je één, drie of zeven dagen vooruit springen. In de grafiek zie je het aantal zieke personen.

Installeer de simulatie

Zet de twee php files in één directory in je document root en test de simulatie uit.

MVC

MVC is een principe waarin je bepaalde zaken scheidt. M staat voor **Model**, dat is de database of eigenlijk de opslag en toegang tot data. De V staat voor **View,** dat is eenvoudig gezegd de GUI, wat de gebruiker ziet: daar zit dus vaak veel HTML en JavaScript bij. Later zullen we ook libraries als bootstrap gaan gebruiken voor de GUI. De C tenslotte, staat voor **Control** en dat is alle logica, zeg maar de 'brains' van het systeem.

Opdracht 1

Beschrijf van de hieronder genoemde onderdelen van de software of ze bij M, V of C horen.

(a) Regel 141 t/m 145

(b) De functie *printPopulation* (regel 57-77)

(c) De functie *passTime* (regel 41-55)

(d) Regel 214 t/m 230 (de table)

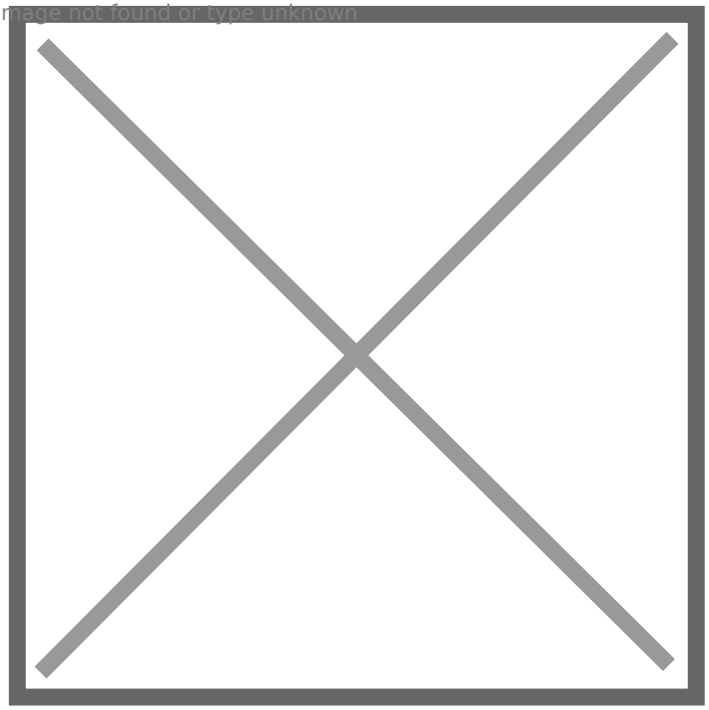
Opdracht 2

De knop "Next 7 >>>" die je in de screen afdruk ziet springt een week vooruit in de simulatie. Deze knop is nog niet geïmplementeerd. Pas de code aan zodat je deze knop wel hebt en zorg ervoor dat de knop ook werkt.

Opdracht 3

In de 'grid' staat elke oranje s voor een ziek persoon, een groene o staat voor iemand die immuun is geworden, - staat voor iemand die nog niet besmet is en ene rode X staat voor iemand die is overleden.

Image not found or type unknown



Verander de code zodat iemand die niet is besmet niet wordt weergegeven, zie voorbeeld hierboven.

Opdracht 4

We gaan functionaliteit uitbreiden, zodat de simulatie-parameters kunnen worden aangepast.

Op regel 3 t/m 8 worden de parameters van de simulatie gedefinieerd.

(a) Zet deze definities in een aparte file, parameters.php en `include 'parameters.php'` in de corona.php file.

(b) Maak een form (editparams.php) om de parameters aan te passen. In dit form worden de waardes uit de parameters.php ingelezen (met een include). Deze parameters worden gebruikt als default values voor het form. Als je het form post dan wordt er een nieuwe parameters.php file geschreven. Hierin staan de aangepaste waarden. Na het aanpassen van de waarden wordt de simulatie (corona.php) vanzelf aangeroepen.

(c) Maak een extra knop op de corona.php pagina die je kunt gebruiken om naar de editparams.php pagina te gaan.

(d) Maak een extra knop op de editparams.php. Indien je deze knop indrukt worden alle waarden in het form weer op de standaard waarden ingesteld. De standaard waarden zijn de waarde zoals ze in de code staan:

```
population=4200;
$popWidth=120;
$contactPerPerson=10; // number of contacts per person
$changeToGetSick=5;   // when you contact someone change to get infected
$changeToDie=3;       // when you are infected, change to die
$daysSick=10;         // days sick
```

Opdracht 5

Pas de code aan, zodat de simulatie begint met 100 mensen die immune zijn.

Pas de editparams.php aan, zodat dit aantal immuun personen kan worden aangepast.

--

Lotto Spel *

Dit is een vrije opdracht waar je meerdere technieken die je hebt geleerd kan gebruiken. Wat je in ieder geval nodig hebt, is HTML Forms en PHP. Met Javascript (of Bootstrap) kun je de interface opleuken.

Lotto

Lotto is een gokspel in Nederland - in principe verlies je hier altijd geld mee. Maar om een gevoel te krijgen hoe het werkt gaan wij het lotto spel naspelen.

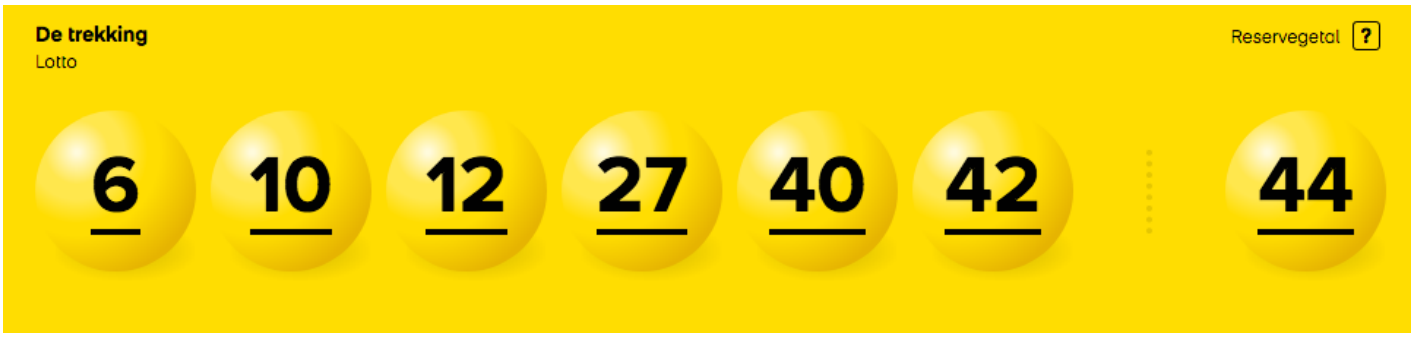
Dit is een vrije opdracht die je kunt gebruiken voor je portfolio.

Wat je moet maken is een manier waarop de gebruiker kan invoeren welke getallen hij kiest. Dan wordt het spel gestart en volgt de trekking. Je laat zien wat de getallen zijn die de gebruiker heeft ingevoerd en hoe hoog de eventuele geldprijs is.

Spelen

Een lot kost 2 euro. Je kiest zelf 6 getallen. Je kunt zes getallen kiezen tussen 1 en 45.

Trekking



Prijzen

Getallen goed	Prijzen
6 getallen goed	Jackpot € 2.500.000,-

Getallen goed	Prijzen
5 getallen + reservegetal goed	€ 25.000,-
5 getallen goed	€ 1.000,-
4 getallen + reservegetal goed	€ 50,-
4 getallen goed	€ 20,-
3 getallen + reservegetal goed	€ 10,-
3 getallen goed	€ 7,50,-
2 getallen + reservegetal goed	€ 5,-
2 getallen goed	€ 2,- speltegoed

Minimale versie

De minimale versie heeft een formulier (form) waar je de nummers in kan opgeven. Je drukt op de knop speel en je simuleert een trekking. Je toont de uitslag, deze bestaat minimaal uit een overzicht met de door jouw gekozen nummers, de getrokken nummers en de geldprijs.

Opties

Maak een handige manier van getallen invoeren, bijvoorbeeld met check boxes. Als je dit doet, zou het natuurlijk mooi zijn dat je niet meer dan 6 getallen kan invoeren.

Na het spelen van het spel wil je misschien nog een keer met dezelfde getallen spelen, maar het makkelijk om nog een keer met dezelfde getallen te 'spelen'.

Houd bij hoe vaak je het spel speelt - of houd bij wat je virtuele banksaldo is. Begin met bijvoorbeeld 100 virtuele euro's.

Patiëntgegevens in JSON File **

In deze opdracht ga je een JSON-file analyseren. JSON is een file formaat om data op te slaan. De data wordt gestructureerd opgeslagen en is leesbaar (als txt). Het formaat lijkt een beetje op een assiociative array en wordt veel gebruikt bij REST API's.

De opdracht

Van een onderzoeksbureau heb je twee JSON-bestanden gehad. Het bestand bevat gegevens van Corona patiënten. Omwille van de privacy van de patiënten worden geen namen gebruikt, maar anonieme patiëntnummers.

Aan elke patiënt wordt een codering toegekend. In het schema hieronder worden de coderingen toegelicht.

Codering	Betekenis
0	Vermoedelijk Covid-19
1	Covid-19 vastgesteld - patiënt ligt op gewone afdeling
2	Covid-19 vastgesteld - patiënt ligt op de intensive care
3	Patiënt is genezen
4	Patiënt is overleden

Je krijgt een functie waarmee je het JSON-bestand kan inlezen. De functie krijgt de file naam mee als parameter en de return value is een assiociative array. Het array ziet er bijvoorbeeld als volgt uit:

Key	Value	Betekenis
p-nc-199134	3	Patiënt p-nc-199134 is genezen
p-nc-199114	2	Patiënt p-nc-199114 ligt op IC
p-nc-199126	0	Patiënt p-nc-199126 heeft vermoedelijk Covid-19

Je krijgt twee bestanden:

p-nc-191.json	bestand van een dag
p-nc-192.json	bestand van de volgende dag

De bestanden staan op deze pagina - links.

Met de functie die hieronder staat kan je het JSON-bestand lezen en krijg je een assiociative array.

```
<?php


function getData($file) { // JSON bestand
  if ( file_exists($file) ) {
    return(json_decode(file_get_contents($file), true)); // return assiociative array
  } else {
    return(0);
  }
}
```

Lees met behulp van de gegeven functie de bestanden in.

De vragen

De opdrachtgever heeft de volgende vragen.

Vraag 1, overzicht

Maak een overzicht van **beide** dagen waarbij van elke categorie wordt aangegeven hoeveel patiënten er zijn.
Bijvoorbeeld:
Er zijn 212 patiënten die vermoedelijk Covid-19 hebben.
Er zijn 376 patiënten waarbij Covid-19 is vastgesteld.
etc.
(het liefst heeft de opdrachtgever deze gegevens netjes in een tabel)

Vraag 2, fouten vinden

Het blijkt dat er bij een paar patiënten een fout is gemaakt bij de invoer van hun status. Hun status is dus niet gelijk aan 1,2,3 of 4. Laat zien om welke patiënten dit gaat en wat de foutief ingevoerde code is. Heb je misschien een idee wat de juiste code zou moeten zijn? Leg uit.

Vraag 3, fouten corrigeren

Met de volgende functie kun je een assiosiative array wegschrijven als JSON-file

```
file_put_contents("file_name",json_encode($array))
```

Kun je de fouten die je bij (2) hebt gevonden corrigeren en het bestand wegschrijven. Doe dit voor beide bestanden.

Vraag 4a, veranderingen

Er zijn op de tweede dag helaas drie patiënten overleden. Zoek op welke patiënten dit zijn (patiëntnummers).

Vraag 4b, veranderingen

Wat was de vorige status van de drie overleden patiënten, komen ze van de IC of wat was hun vorige status?

--

Quizzes