

# Zadania Backend Developer

**Ważne!** Zrób tyle ile zdołasz. Jeżeli nie uda się wykonać wszystkiego, nie przejmuj się, wyślij tyle ile udało się zrobić.

## PHP

### 1.) Zadanie wymagane - Łamacz kodu

Mając klucz rozszyfrowujący, stwórz program który odkoduje poniższą, zaszyfrowaną wiadomość.

Każdemu znakowi z klucza przyporządkowany jest znak z alfabetu za pomocą którego powinien zostać odszyfrowany. Np.: za pomocą znaku @ została zaszyfrowana litera **k**. Jeżeli danego znaku nie ma w kluczu to powinien on znaleźć się w odszyfrowanej wiadomości w niezmienionej formie.

Klucz z przypisanym alfabetem:

Plain text

```
1  ! ) " ( £ * % & > < @ a b c d e f g h i j k l m n o
2  a b c d e f g h i j k l m n o p q r s t u v w x y z
```

Zaszyfrowana wiadomość:

Plain text

```
1  )g!ld, j(!ad "> h>£ gdol>!o!" o!(!c>£
```

Zadanie dodatkowe. Rozbuduj program o możliwość szyfrowania. Zaszyfruj poniższą wiadomość a później przepuść ją przez deszyfrator i sprawdź czy wygląda ona tak samo.

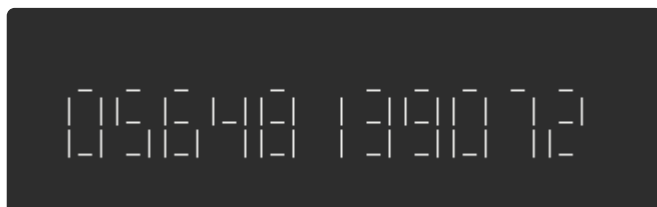
JavaScript

```
1  Zażółć, gęślą jaźń.
```

## 2.) Zadanie opcjonalne - Wyświetlacz LCD

Napisz program który mając podaną liczbę wypisze ją na ekran w formacie podobnym do tego używanego w wyświetlaczach LCD.

Przykładowo, podając liczbę **05648139072** chcemy uzyskać następujący wynik w konsoli:



Cyfry do użycia:

Plain text

1	-	-	-	-	-	-	-	-
2				-				-
3		-			-			-

## SQL:

Używając struktury:

### SQL

```
1  -- Loterie
2  create table `lotteries` (
3      -- Id
4      `id` int unsigned not null,
5      -- Nazwa loterii
6      `name` varchar(255) not null,
7      -- Cena za los
8      `ticket_price` decimal(12, 2) not null,
9      -- Wysokość nagrody pieniężnej
10     `prize` decimal(12, 2) not null,
11     primary key (`id`)
12 );
13
14 -- Losowania
15 create table `draws` (
16     -- Id
17     `id` int unsigned not null,
18     -- Id loterii
19     `lottery_id` int unsigned not null,
20     -- Data losowania
21     `draw_date` timestamp not null,
22     -- Wygrany numer
23     `won_number` int,
24     primary key (`id`),
25     constraint `draws_lottery_id`
26         foreign key (`lottery_id`)
27         references `lotteries` (`id`)
28         on delete cascade on update cascade
29 );
30
31 -- Losy zakupione przez uzytkownikow
32 create table `tickets` (
33     -- Id
```

```

34         `id` int unsigned not null,
35         -- Id losowania
36         `draw_id` int unsigned not null,
37         -- Data zakupu losu
38         `bought_date` timestamp not null,
39         -- Wytypowany numer
40         `number` int unsigned not null,
41         primary key (`id`),
42         constraint `tickets_draw_id`
43             foreign key (`draw_id`)
44             references `draws` (`id`)
45             on delete cascade on update cascade
46     );

```

Oraz danych:

#### SQL

```

1  insert into `lotteries` values
2  (1, 'GG World Million', 10.49, 300.00),
3  (2, 'GG World X', 12.99, 400.00);
4
5  insert into `draws` values
6  (1, 1, '2021-06-15 14:00:00', 32),
7  (2, 1, '2021-07-15 14:00:00', 47),
8  (3, 1, '2021-08-15 14:00:00', 5),
9  (4, 1, '2021-09-15 14:00:00', null),
10 (5, 2, '2021-07-01 16:00:00', 29),
11 (6, 2, '2021-08-01 16:00:00', 4),
12 (7, 2, '2021-09-01 16:00:00', null);
13
14 insert into `tickets` values
15 ( 1, 1, '2021-05-16 18:21:34', 21),
16 ( 2, 1, '2021-06-02 13:31:02', 29),
17 ( 3, 1, '2021-06-15 14:00:02', 13),
18 ( 4, 2, '2021-06-30 03:44:34', 47),
19 ( 5, 2, '2021-07-02 07:09:02', 32),
20 ( 6, 2, '2021-07-15 14:00:15', 5),
21 ( 7, 3, '2021-07-15 15:44:34', 13),
22 ( 8, 3, '2021-07-28 04:29:11', 5),

```

```
23  ( 9, 3, '2021-08-15 13:59:58', 5),
24  (10, 3, '2021-08-15 14:00:01', 5),
25  (11, 4, '2021-08-25 01:15:48', 38),
26  (12, 5, '2021-06-04 22:01:53', 29),
27  (13, 5, '2021-06-24 05:25:09', 13),
28  (14, 5, '2021-07-01 16:00:05', 29),
29  (15, 5, '2021-07-01 16:00:02', 24),
30  (16, 5, '2021-07-01 16:00:03', 11),
31  (17, 6, '2021-07-24 04:32:56', 4),
32  (18, 6, '2021-08-01 16:02:01', 13),
33  (19, 7, '2021-07-16 22:34:49', 36),
34  (20, 7, '2021-08-04 16:00:55', 15),
35  (21, 7, '2021-08-04 16:00:55', 49),
36  (22, 7, '2021-08-04 16:00:55', 46);
```

Wyobraź sobie, że pracujesz nad systemem odpowiedzialnym za sprzedaż losów na loterię. Użytkownicy mogą kupować losy w których typują pojedynczą liczbę z przedziału 1-49. Losy mogą zakupić na jedną z dwóch loterii. Loterie różnią się ceną jednostkową za los, częstotliwością losowań oraz wysokością nagrody pieniężnej. Użytkownik kupując los, kupuje go zawsze na najbliższe losowanie które dopiero ma się odbyć. Kiedy przychodzi moment losowania, automat generuje liczbę z założonego przedziału i wpisuje ją w odpowiednie pole rekordu, tego losowania. Po zakończeniu całej operacji, system dodaje rekord z nowym losowaniem, zawierający kolejną, przyszłą datę. W tym momencie użytkownicy ponownie otrzymują możliwość zakupu losów na tę loterię. Wiążąc wygenerowaną przez automat liczbę z liczbami wytypowanymi przez użytkowników i wypisanymi na losach, można wyłonić zwycięzców, którzy powinni otrzymać nagrody.

## 1.) Zadanie wymagane - Wygrane losy

Napisz zapytanie które zwróci listę zawierającą identyfikatory wszystkich losów, które okazały się wygrane dla loterii "GG World X".

## 2.) Zadanie opcjonalne - Statystyka zarobków

System czasami działa wadliwie i zdarza się, że użytkownikom udaje się kupić los już po tym jak losowanie się odbyło. W takim przypadku, los uznajemy za nieważny i musimy oddać użytkownikowi pieniądze które za niego zapłacił.

Napisz zapytanie które podsumuje przychód z losów (sumę wszystkich zakupionych losów) oraz realny zysk (przychód po odjęciu sumy za nieważne losy) za lipiec 2021. Zapytanie powinno zwrócić listę z następującymi wartościami: nazwa loterii, przychód, zysk.

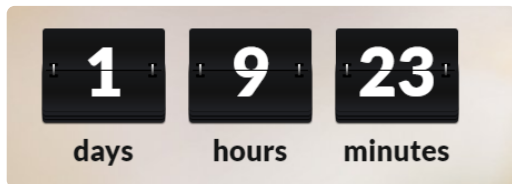
---

# Frontend

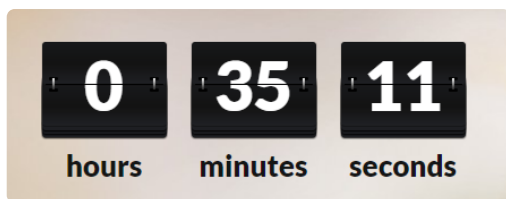
## 1.) Zadanie opcjonalne - Licznik

Stwórz licznik który odmierza czas do przeprowadzania losowania loterii. Licznik powinien różnie wyglądać w zależności od tego ile pozostało do losowania.

Gdy pozostało więcej niż jeden dzień, powinien wyświetlać: dni, godziny, minuty.



Gdy pozostało mniej niż jeden dzień, powinien wyświetlać: godziny, minuty, sekundy.



Ostyluj go by wyglądał tak jak na obrazku powyżej.

Możesz wykorzystać grafikę z załącznika.

### Wykorzystaj technologie:

HTML,

CSS wraz SCSS,

BOOTSTRAP 4,

JQUERY