

Solutions

version 1.1.0



Mythologie Grecque

I. Lucile amour et rock'n roll

Le but de cet exercice est de comprendre le fonctionnement du programme. Le langage est le [Rockstar](#). La solution qu'on attend est d'affecter 1334 à "my voice" de plusieurs façons (soit en chiffre soit avec 'a big bad wolf', par exemple). Le challenge peut aussi être résolu en affectant 1 à "my voice" (un modulo par 1 donne toujours 0). Les participants peuvent aussi lire la solution directement dans le code, puisqu'il suffit de traduire les mots en chiffres.

Il n'y a pas qu'une seule solution à ce challenge ; vous pouvez inviter les participants à chercher les autres

Le petit chef

La première étape est de traduire la recette. Celle-ci est en Greeklish un mélange d'anglais et de grec. Il y a un traducteur en ligne greeklish vers anglais « <http://speech.ilsp.gr/greeklish/> » Ce qui donne :

```
ελληνική σαλάτα  
5 ντομάτες  
1 ΦΕΤΑ η 200 γραμμάρια  
2,5 κόκκινα κρεμμύδια  
6 κουταλάκια του γλυκού ελαιόλαδο  
4 κουταλάκια του γλυκού ξύδι  
2,5 κουταλάκι του γλυκού ρίγανη  
  
ΣΗΜΑΝΤΙΚΟ: Ο Dionysus τρώει για 75
```

Une fois que nous avons la recette en grec on doit la traduire en français sur google translate par exemple. Ce qui donne :

```
Salade grecque  
5 tomates  
1 FETA ou 200 grammes  
1 concombre  
2.5 oignons rouges  
6 cuillères à café d'huile d'olive  
4 cuillères à café de vinaigre  
2.5 cuillère à café d'origan  
  
IMPORTANT: Dionysus mange pour 75
```

Une fois que nous avons la recette en français il faut l'appliquer dans le langage de programmation ésotérique « chef ». Il faut bien évidemment multiplier les quantités par 75 ce qui donne :

```
Grec Salad.  
Ingredients.  
375 tomatoes  
15000 g feta  
75 concombres  
187 oignons rouges  
450 teaspoons huile d'olive  
300 teaspoons Vinaigre  
187 teaspoons Origan  
Method.  
Put tomatoes into mixing bowl. Put feta into mixing bowl. Put concombres  
into mixing bowl. Put oignons rouges into mixing bowl. Put huile  
d'olive into mixing bowl. Put Vinaigre into mixing bowl. Put Origan  
into mixing bowl. Liquify contents of the mixing bowl. Pour contents  
of the mixing bowl into the baking dish.  
Serves 1.
```

II. Le fou du volant

Il faut implémenter un algorithme qui donne la somme des entiers multiples de 3 et ou de 5 inférieurs à 3000.



Harry Potter

Une Bombe à Gringot

Une solution possible :

```
// On ajoute une méthode aux Strings pour remplacer un char à
une position
String.prototype.replaceAt = function(index, replacement) {
    return this.substr(0, index) + replacement +
this.substr(index + replacement.length);
}

// on import le fichier encrypté qui génère le code de la
bombe
const brute = require("./findTheCode.js");

// on défini la size du code, l'alphabet utilisé pour générer
le code
// on génère une String de base sur laquelle on va travailler
(ici AAA) et le résultat qu'on attend du fichier ecrypté
let codeLength = 3;
let alphabet =
"ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz01234567
89+~/*^\\!?:,%_()[]{}&";
let code = generateBaseString();

let res = {success: false, message: ""};

// pas le plus optimisé mais ça fonctionne pour une String de
taille 3, il suffit d'ajouter un bloc par nombre de fois où
on veut aller plus loin dans une String
for(let pos = 0; pos < codeLength; pos++) {
    for(let letter = 0; letter < alphabet.length; letter++)
    {
        for(let pos2 = pos + 1; pos2 < codeLength; pos2++)
        {
            for(let letter2 = 0; letter2 <
alphabet.length; letter2++) {
                for(let pos3 = pos2 + 1; pos3 <
codeLength; pos3++) {
                    for(let letter3 = 0; letter3 <
alphabet.length; letter3++) {
                        code = code.replaceAt(pos3,
alphabet.charAt(letter3));
                        console.log(code);
                        res = brute(code);
                        if (res.success) {
                            console.log(`Code:
${code}\nFlag: ${res.message}`);
                        }
                    }
                }
            }
        }
    }
}
```

```

        return;
    }
}
    }
    code = code.replaceAt(pos2,
alphabet.charAt(letter2));
    console.log(code);
    res = brute(code);
    if (res.success) {
        console.log(`Code: ${code}\nFlag:
${res.message}`);
        return;
    }
}
    }
    code = code.replaceAt(pos,
alphabet.charAt(letter));
    console.log(code);
    res = brute(code);
    if (res.success) {
        console.log(`Code: ${code}\nFlag:
${res.message}`);
        return;
    }
}
}

// on génère une String de taille codeLength avec la première
lettre de l'alphabet alphabet
function generateBaseString() {
    let result = "";
    for (let i = 0; i < codeLength; i++)
        result += alphabet[0];
    return result;
}

```

Le token est : *Cannelle et Fleur de Sucre sont les meilleurs*

Le Fourchelang

Pour ce challenge, les participants doivent bien délimiter les trois textes en se servant des guillemets par exemple. Le bon texte à « traduire » est le deuxième.

```
std::string Fourchelang::Translator()
{
    char firstArray[] = {'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', '\0'};
    std::string secondArray[] = {"A", "S", "S", "R", "E", "R",
        "H", "A", "I", "A", "A", "SR", "SER", "H", "AE", "SAR", "SS",
        "EE", "RR", "SHH", "AI", "HH", "RR", "AA", "AAH", "SH",
        "\0"};
    std::string data = "";
    std::string fourchelang = "";

    std::ifstream inputFileStream("vieuxLivre.txt");
    std::getline(inputFileStream, data);
    std::cout << data << std::endl;
    //Ajoutez une fonction qui chercher les ' ' ' pour séparer
    les textes
    for (std::string::size_type count = 0; count <
        data.length(); ++count)
    {
        char translate = data[count];
        for (int index = 0; firstArray[index] != '\0'; index++)
        {
            if (std::tolower(translate) == firstArray[index]) {
                fourchelang = fourchelang + secondArray[index];
                break;
            }
            else if (firstArray[index + 1] == '\0') {
                fourchelang = fourchelang + translate;
            }
        }
    }
    std::cout << fourchelang << std::endl;
    return data;
}
```

Le token est : *AAEEEEAAH HSRIRRRRA SRE SRAEHH RAI SERAIEE. ISR
ESERSARAEIHHA SRE SEEAESAESHH SSAII EEéSARAHRAISHH SRE HHEHIH
RAHRR RRAEH RRAHH ESHH SR'AEEEEASAA RE RRAEH SEEARR. SERAIRR
ISR RRAHHAISHH SSAI'ISR éSHHAISHH SHHEEAESAR SHHAEER. AIHE
RAEAISREAIEE SEEûSRAHSHHE RRE RIRRAIRRAISHH SREHSHHESEREHSHH*

RAHRR RRAEH SAEESARRR. ISR SRAIRRRRA SHHAESERSEEE SRE
SEEAESAESHH SEEIRRé RAI RREESAREHSHH ESHH EEEHAEERA RRAEH
SAREEAESAREEE RRAHH ISERSAREEéHHEEE SR'éSHHAERRE RE RRA
EEAESE. RRA HHIRRIA EH RRE SEEAEAIISRSRA, SRA SAASERSEEE RERR
RRESEESHRR RRE RAEHRAISHH EH AIH SHHAEIEESISRSRAEH RE
SAEISREAIEERR SHHEEHERR !

Affronter le Basilic

Il n'y a pas d'algorithme particulier attendu pour ce challenge. Le but est surtout de battre de Basilic même s'il faut relancer le script beaucoup de fois. Attention cependant à ce que les participants ne touchent pas aux différents fichiers. Ils n'ont droit que le droit de modifier la méthode `chooseAttack()` de la class Aventuriers, se trouvant dans le fichier éponyme.

Le token est : *Henri Podbeurre et la coupe de cheveux*

Tron

L'arrivée dans Tron

Au tout début, il faut décoder les phrases codées en Ascii ensuite modifier les fichiers.

Flag : M@TRIX BL@STER

Phrase 1 :

Le fichier à modifier c'est Make_Board

Phrase 2 :

Le fichier à modifier c'est Make_Surf

Phrase 3 :

Le fichier à modifier c'est Game

Phrase 4 :

Le fichier à modifier c'est Board

Phrase 5 :

Le fichier à modifier c'est SubSurf

Make Board

```
def make_board(image_path, board_rect):
    x, y, width, height = board_rect

    puzzle_image = Make_Surf.load_puzzle_image(image_path, board_size=(width,height))
    subsurfs = list(Make_Surf.make_subsurfaces(puzzle_image, offset=(x, y)))

    return Board.Board(x, y, subsurfs[:-1])

def random_moves(n):
    directions = ('up', 'right', 'down', 'left')
    for _ in range(n):
        yield choice(directions)
```

Dans ce fichier le participant doit retrouver le nom de la fonction manquante et aussi les mots manquants sur la variables directions.

Make Surf

```
def load_puzzle_image(path, board_size):
    surface = pygame.image.load(path).convert_alpha()
    return pygame.transform.scale(surface, board_size)

def make_subsurfaces(surface, offset=(0, 0)):
    width, height = surface.get_size()

    assert width % 3 + height % 3 == 0, (
        "la dimension de l'image n'est pas divisée par 3: {}".format(surface.get_size()))

    offx, offy = offset
    for y in range(0, height, height//3):
        for x in range(0, width, width//3):
            subsurface = surface.subsurface(x, y, width//3, height//3)
            yield SubSurf.SubSurf(offx + x, offy + y, subsurface)
```

Dans cette partie il manque une variable qui permet d'avoir l'échelle de l'écran et aussi les chiffres ne sont pas corrects.

Game

```
def update(self, elapsed):
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            self.quit()

        elif (event.type == pygame.KEYDOWN
              and event.key == pygame.K_UP):
            self.board.move('up')

        elif (event.type == pygame.KEYDOWN
              and event.key == pygame.K_RIGHT):
            self.board.move('right')

        elif (event.type == pygame.KEYDOWN
              and event.key == pygame.K_DOWN):
            self.board.move('down')

        elif (event.type == pygame.KEYDOWN
              and event.key == pygame.K_LEFT):
            self.board.move('left')
```

Dans cette partie vous devez rajouter les events des touches right, down et left.

Board

```
x, y = {
    'up': (0, -y_spacing),
    'right': (x_spacing, 0),
    'down': (0, y_spacing),
    'left': (-x_spacing, 0)
}[direction]
```

Dans cette partie, il manque des variables qui permette de gérer les différentes directions des tuiles.

SubSurf

```
@property
def height(self):
    return self.image.get_height()

@property
def rect(self):
    return pygame.Rect(
        self.x,
        self.y,
        self.width,
        self.height
    )
```

Dans cette partie il manque le return de la fonction **height** pour avoir la hauteur de l'image et dans la fonction **rect** il manque les variables y et height.

Le créateur

Il faut utiliser un logiciel pour décrypter le son.

La face caché d'ITAKALA

Login:

Le login est X_Legacy

Pour le login vous aurez une phrase en ascii à décoder.

Mot de passe :

STEVENLISBERGER

Pour le mot de passe vous aurez une audio comme dans le 2eme challenge qu'il faudra décrypter.

Solution: curl -- data "username=X_Legacy&password=STEVENLISBERGER"
54.38.232.200:4242/login.php

Fallout

Challenge N°1

Le résultat du flag final est : 3{7|JdtYrQ6m|oY4U!5J(r8ztZ&2Gu6(

Pour arriver à cela, l'utilisateur devra comprendre les phrases du sujet.

Quand mon père n'est pas là, je suis là pour aider les gens à mieux comprendre la situation :
Les alt sont les descriptions des images lorsqu'elles ne chargent pas. Donc si le papa = img n'est pas là, le texte sera là pour donner des informations à sa place.

Je suis le lien qui guide les hommes vers de nouveaux chemins :
Après avoir cliqué sur le bouton register, l'url de la page est le flag. L'url guide les gens sur internet.

Souvent dans le noir, j'aime me cacher. Mon premier est le signe d'une grande destinée :
J'aime me cacher dans le noir car texte écrit en noir sur du noir et mon premier car le premier caractère de chaque ligne est le flag.

Je suis le dieu de cette nouvelle ère et tout commentaire à mon égard sera sanctionné :
Le nouveau dieu = itakala et tout commentaire à mon égard sera sanctionné" car # en sql est un commentaire.

3{7|JdtY -> Lorsque vous ouvrez le code source de la page, vous verrez dans les balises img avec un alt="string", toutes les images ont un flag de 9 caractères sauf 1.

```
<html>
  <head>...</head>
  <body id="page-top"> == $0
    <nav class="navbar navbar-light navbar-expand-lg fixed-top bg-secondary text-uppercase" id="mainNav">...</nav>
    <header class="masthead text-white text-center">...</header>
    <div class="container-fluid">
      <div class="row justify-content-between">...</div>
      <div class="row justify-content-between">
        <div class="col-4">...</div>
        <div>...</div>
        <div class="col-4">
          
        </div>
      </div>
      <div class="copyright py-4 text-center text-white">
        <div class="container">
          <small>Copyright © Coding Club 2020</small>
        </div>
      </div>
    </body>
  </html>
```


rQ6m|oY4 -> Cliquez sur le bouton register du site. Dans l'url, le nom du fichier rQ6m%7CoY4.php est le mini flag. Il faut juste url décode.

54.38.232.200:4444/rQ6m%7CoY4.php

Decode from URL encoded format


Simply enter your data then push the decode button.

rQ6m%7CoY4

 For encoded binaries (like images, documents, etc.) use the file upload form a bit further down on this page.

UTF-8  Source character set.

☐ Decode each line separately (useful for multiple entries).

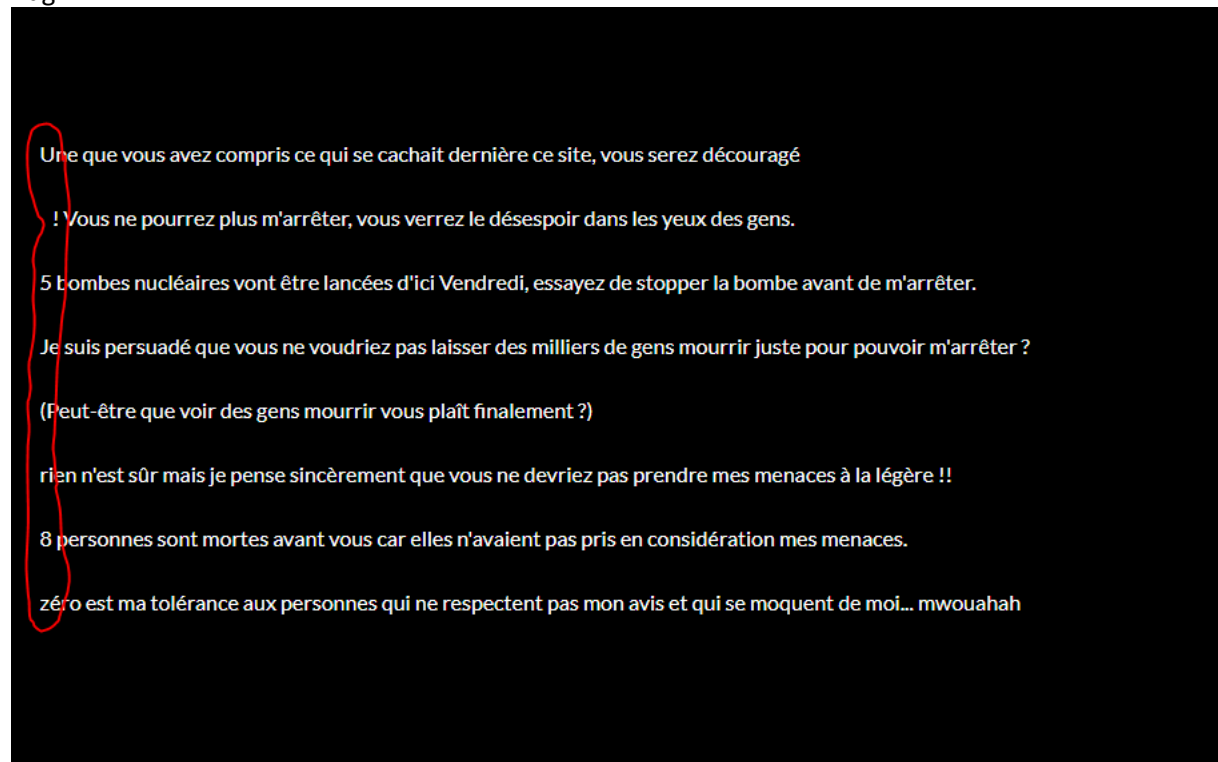
 Live mode OFF Decodes in real-time when you type or paste (supports only UTF-8 character set).

< DECODE >

Decodes your data into the textarea below.

rQ6m|oY4

U!5J(r8z -> Un texte est écrit en noir dans index.php. Chaque première lettre de chaque ligne est le flag



tZ&2Gu6(-> La requête sql à réaliser est itakla'#, elle vous affichera le mot de passe du compte qui est le flag.

itakala'#'	password	login
------------	----------	-------

```
array(3) { ["id"]=> string(1) "5" ["username"]=> string(7) "itakala" ["password"]=> string(8) "tZ&2Gu6(" }
```

Challenge N°2

Initialisation :

```
import sys
import math

# Lire le nombre d'habitant par clan et en déduire leur besoin en eau soit 2 litre par personne
clan1 = int(sys.argv[1]) * 2
clan2 = int(sys.argv[2]) * 2

# Trouver le besoin total des deux clans
need = clan1 + clan2

# Initialiser le ratio de chaque clan pour un voyage au puit
ratio_clan1 = 0
ratio_clan2 = 0
```

On récupère les arguments du programme et on initialise toutes les variables nécessaires au partage et la priorité des ressources.

Vérification :

```
# figure out the ratio of water needed for each clan
# using the smallest clan for proportion
if (clan1 >= clan2):
    # getting the proportion of water needed from the smallest clan in relation to the whole population
    ratio = need / clan2
    # clan 2 must be handled first so we round it's ratio up
    ratio_clan2 = math.ceil(10 / ratio)
    # clan 1 gets the size of the bucket minus the clan 2's share
    ratio_clan1 = 10 - ratio_clan2
else:
    # getting the proportion of water needed from the smallest clan in relation to the whole population
    ratio = need / clan1
    # clan 2 must be handled first so we round clan 1's ratio down
    ratio_clan1 = math.floor(10 / ratio) if (10 / ratio > 1) else 1
    # clan 2 gets the size of the bucket minus the clan 1's share
    ratio_clan2 = 10 - ratio_clan1

print("ratio clan 1 : " + str(ratio_clan1) + "; ratio clan 2 : " + str(ratio_clan2))

# initiate the bucket
bucket = 0
```

On compare les valeurs pour savoir quelle proportion réaliser dans le partage d'eau.

Boucle :

```
while (clan1 > 0 or clan2 > 0):
    # fill the bucket
    line = "f"
    bucket = 10

    # if clan2 still needs more than it's ratio
    if (clan2 > ratio_clan2):
        # give clan2 its ratio
        line += "g" + str(ratio_clan2)
        bucket -= ratio_clan2
        clan2 -= ratio_clan2
    # if clan2 need less than their ratio
    elif (clan2 > 0):
        # give clan2 what they need left
        line += "g" + str(clan2)
        bucket -= clan2
        clan2 = 0

    # if clan2 has all their water
    if (clan2 == 0 and clan1 >= bucket):
        # deal clan1 the whole bucket
        line += "d" + str(bucket)
        clan1 -= bucket
        bucket = 0
    # if clan2 still needs water & clan1 still needs water
    elif (clan1 > ratio_clan1):
        # deal clan1 their ratio
        line += "d" + str(ratio_clan1)
        clan1 -= ratio_clan1
        bucket -= ratio_clan1
    # if clan 1 need less than their ratio
    elif (clan1 > 0):
        # give them what they need left
        line += "d" + str(clan1)
        bucket -= clan1
        clan1 = 0

    # print the current
    print(line, end='')
# print the rest of the water in the bucket
print("r" + str(bucket))
```

Mais comment on en est arrivé là ?


```

# a message to start with
starting = "Bonjour, je suis un des présidents des états unis !"
# a message to end with
ending = "ciao"
# Random messages to simulate a conversation
answers = [
    "Je n'ai pas le temps pour ces balivernes j'ai des responsabilités !",
    "Hmm, intéressant je me pencherais sur le sujet plus tard.",
    "Laissez moi avec cette affaire !",
    "Je suppose que je peux y réfléchir",
]

# line is used to stop the bot when someone says "ciao"
line = "Bonjour"
# prints the first line
puts(starting)
# keep speaking as long as the conversation isn't over
while line != "ciao" do
    # get the user input
    line = gets().chomp()
    # if the input is "ciao" stop the conversation
    # else print a random answer
    if line == "ciao"
        puts ending
    else
        puts(answers[rand(0..3)])
    end
end
end

```

Attention au cerveau !

```
import sys

if (len(sys.argv) != 2):
    print("Send the crypted file name as an argument")
    exit(1)

file = open(sys.argv[1], "r")
code = file.read()

memory = [0]
n = 0
i = 0

while i < len(code):
    char = code[i]
    if (char == 'd'):
        memory[n] += 1
    elif (char == 'r'):
        memory[n] -= 1
    elif (char == 'u'):
        print(chr(memory[n]), end='')
    elif (char == 'm'):
        n += 1
        if (len(memory) <= n):
            memory.append(0)
    elif (char == 'l'):
        n -= 1
    elif (char == 'e'):
        if (memory[0] == 0):
            while code[i] is not 's':
                i += 1
    elif (char == 's'):
        if (memory[0] != 0):
            while code[i] is not 'e':
                i -= 1
    i += 1
```