

## **Mini-projets :**

# **la chasse aux palindromes**

Lycée Charles Gide

Rapport de projet informatique de terminale scientifique,  
spécialité ISN  
du 01/11/13 au 21/11/13

**CARBON Dimitri TS2**  
**CHABALIER Andy TS2**

# Remerciements

Nous tenons tout d'abord à remercier M. GROTTI

pour son aide dans l'élaboration de ce mini projet.

Nous remercions aussi les membres du site [www.developpez.net](http://www.developpez.net) pour la documentation pour optimiser notre programme.

# Table des Matières

1. Introduction.....	1
1.1 Généralités.....	1
1.2 Le sujet.....	1
1.3 Cahier des charges.....	1
2. Organisation du projet.....	5
2.1 Organisation du travail.....	5
2.2 Choix des outils de développement.....	5
3. Analyse préalable du projet.....	9
4. Analyse technique (Développement).....	13
5. Manuel d'utilisation.....	15
6. Perspectives et conclusions.....	17
6.1 Perspectives.....	17
6.2 Conclusions.....	17
6.2.1 Fonctionnement de l'application.....	17
6.2.2 Fonctionnement du groupe de travail.....	17
7. Annexe : Listings identifiés et commentés.....	21

# **1. Introduction**

## **1.1 Généralités**

Le mini-projet de programmation nous permet de nous entraîner à la réalisation d'un programme, ainsi que d'un compte rendu de projet. D'une durée de 20 jours, l'exercice doit nous permettre de nous aider à comprendre les exigences du bac. Encadré par notre Professeur, chaque couple d'élève doit remettre un programme fonctionnel et un rapport de projet.

## **1.2 Le sujet**

Concevez un script python qui détermine si une phrase donnée est un palindrome,

Vous remarquerez les caractéristiques de ces divers palindromes. Votre programme devra distinguer un cas tel que « À l'étape, épate-la ! » qui est un palindrome si on néglige les accents mais n'en est pas un si l'on tient compte des accents (problème du a), d'un cas d'une phrase qui est un palindrome y compris en tenant compte des accents.

## **1.3 Cahier des charges**

Le programme doit pouvoir déterminer si une phrase ou un mot, est un palindrome (mot ou phrase qui peut se lire indifféremment dans les deux sens).

Le programme doit récupérer une phrase saisie par l'utilisateur. Cette phrase doit ensuite être traitée par ledit-programme pour déterminer si la phrase est ou non un palindrome.

Le programme doit contenir un choix de traitement de la phrase.

\*Premier choix : la phrase est un palindrome sans compter les accents et majuscules-----> suppression des accents ; conversion des majuscules en minuscules

\*second choix : la phrase est un palindrome en comptant les accents et des majuscules.

Le programme doit pouvoir se répéter selon le choix de l'utilisateur pour améliorer l'ergonomie

le programme doit permettre à l'utilisateur de taper sa phrase avec des espaces

## **2. Organisation du projet**

### **2.1 Organisation du travail**

Après avoir réfléchi en commun sur l'analyse du projet, nous nous sommes réparti les tâches. Pendant qu'un de nous s'occupait du codage, l'autre élaboré le dossier. Relié par le logiciel de visioconférence Skype, cette répartition a été peu à peu abolie, pour donner un travail commun. Les séances en classe nous a permis de définir des objectifs à atteindre.

### **2.2 Choix des outils de développement**

Nous avons utilisé IDE de python. Le compilateur et le débogueur étant intégré. Nous avons choisi la version 3.3 de Python, puisque c'est le langage que nous allons devoir utiliser pour le projet final. C'est aussi le langage que nous utilisons en séance.

## 3. Analyse préalable du projet

### 3.1 Analyse du projet

Dans un premier temps, nous avons dû regarder de quelle manière on détermine la présence d'un palindrome ou non. Il suffit de vérifier si un terme  $i$  est égal à son symétrique dans la phrase. Pour cela nous avons posé la formule suivante :

$$P[i]=P[PI-i-1]$$

*P* étant la liste contenant la phrase à tester, et *PI* la taille de la phrase.

Avec ceci, nous avons un programme qui permet de vérifier si une phrase est un palindrome ou non.

Mais nous nous heurtons à plusieurs problèmes. Tout d'abord, l'utilisateur doit entrer sa phrase sans espace. L'espace étant considéré comme un caractère. Nous devons donc chercher un moyen de supprimer les espaces lors du traitement.

Ensuite, pour répondre aux exigences du sujet, nous devons trouver un moyen de faire une vérification de la palindromicité en tenant compte des accents, et un autre, sans les accents.

Nous devons donc chercher un moyen pour supprimer les accents, pour pouvoir distinguer le choix de l'utilisateur (en tenant compte des accents ou non).

Par exemple une conversion de l'unicode en ACSII, qui permettrait de gérer le problème des accents.

Par la même occasion, éliminer ou non les majuscules.

Pour la fonction de refaire une recherche(recommencer) une boucle while doit être utilisée.

## 4. Analyse technique

### 4.1 Fonction et librairie

```
1     import unicodedata #librairie utilisé pour enlever les accents
2     de la chaîne
3
4
5     def palindrome():      #on crée la fonction qui détermine la
6     validité du palindrome
7
8         for i in range(0,len(phrase)):
9             if phrase[i] == phrase[len(phrase)-1-i]:      #on test la
10            i valeur avec celle qui correspond(la dernière valeur-la valeur de
11            i)
12                print("etape",i,"correcte")
13                resultat=1
14            else:
15                resultat=2
16                print("etape",i,"incorrecte")
17                break      #arrêt de la boucle. ce n'est pas un
18            palindrome
19
20        if resultat==1:
21            print("votre phrase est un palindrome")
22        else:
23            print("votre phrase n'est pas un palindrome")
24        return
```

« `import unicodedata` » est une instruction d'appel d'une librairie permettant au cours du programme, de pouvoir utiliser la conversion Unicode--> ACSII, pour pouvoir supprimer les accents d'une phrase.

« `def palindrome():` » est la fonction qui détermine si la phrase est ou non un palindrome. Voyons en détail cette fonction.

« `for i in range(0,len(phrase)):`  
    `if phrase[i] == phrase[len(phrase)-1-i]:` » la fonction vérifie que chaque nombre `i` allant du premier terme, au dernier terme de la phrase (donné par le `len()`) est égal à son symétrique. Ce symétrique est donné par la formule `len(phrase)-1-i`.

### 4.2 Corps du programme

Une boucle `while` englobe tout le programme. Elle permet grâce à une variable initialisé en 'Y' de recommencer le programme ou de le quitter, selon le choix de l'utilisateur.

```
43         phrase = phrase.strip() #élimine les espaces parasite au
44         début et à la fin
45         phrase = phrase.replace(" ", "") #supprime tous les espaces
```

## la chasse aux palindromes

Ces deux instructions permettent de « nettoyer » la phrase. Tout d'abord les espaces qui ont pu s'insérer au début à la fin des phrases. Puis les espaces entre les mots. En effet, sans ce passage, les espaces sont considérés comme des caractères de la phrase, et empêche de faire tourner le programme correctement. On les supprime donc, tout en laissant l'utilisateur les insérer pour son confort.

```
61 #simplification de la phrase au maximum
62
63     phrase = phrase.lower() #enleve les majuscules
64     phrase =
65 unicodedata.normalize('NFKD',phrase).encode('ASCII', 'ignore')
66     #ligne qui permet de convertir de l'Unicode en
67 ASCII(destruction des accents)
```

Ces deux instructions se trouvent dans la partie du programme où on vérifie si la phrase est un palindrome sans tenir compte des accents et des majuscules.

« unicodedata.normalize('NFKD',phrase).encode('ASCII', 'ignore') » est une instruction récupérée à l'aide de la documentation du site [www.developpez.net](http://www.developpez.net). Elle convertit les caractères de la phrase (Unicode) en ASCII, ce qui nous permet de comparer les deux caractères sans la barrière des accents.

On remarque pour cette instruction, l'utilisation de la librairie unicode que nous avons importé au début du programme.



## **5. Manuel d'utilisation**

Après avoir exécuté le programme, un rappel de la définition d'un palindrome apparaît.

Vous devrez choisir une des deux branches du programme :

\*choix 1 : le programme tiendra compte des accents et des majuscules. Par exemple, les mots « ère » et « Ere » seront considérés comme des non-palindromes.

\*choix 2 : le programme ne tiendra pas compte des accents et des majuscules. Dans ce cas-là, les mots « ère » et « Ere » seront considérés comme des palindromes.

Le programme vous demandera ensuite d'entrer votre phrase. (le programme gère les espaces)

Quel que soit votre choix, vous pouvez insérer les accents et majuscules dans votre phrase. Le programme s'occupera lui-même de les ignorer ou non, selon votre choix.

Le programme est lancé. Il vous affichera le nombre d'étape qu'il a réalisé (qui correspondent au nombre de caractère de votre phrase. Vous pouvez donc voir quel est le caractère qui n'a pas de jumeaux.)

Il conclura donc en vous disant la nature de la phrase.

Le programme va ensuite vous demander de choisir de recommencer ou non.

## **6. Perspectives et conclusions**

### **6.1 Perspectives**

Notre programme des palindromes, est dans son état actuel, finit. Mais comme tout programme informatique, il peut être amélioré. Tout d'abord, l'ergonomie. La console (ou le shell) est un environnement sympathique, mais pas aussi ergonomique que peut l'être un programme avec une interface graphique. Une optimisation est toujours possible. Notamment, on peut donner à l'utilisateur plus de choix que les deux proposés, par exemple, un véritable palindrome (accent, majuscules, caractères spéciaux....) un sur-simplifié, seulement avec les majuscules, juste les accents..... et même, avec un dictionnaire de mot, proposer un mot qui permettrait de faire le palindrome lorsque la phrase initiale n'est pas un palindrome.

### **6.2 Conclusions**

#### **6.2.1 Fonctionnement de l'application**

Après avoir testé le programme de diverse façon, en essayant de l'induire en erreur, aucun bogue n'a été révélé, et non corrigé. Les instructions d'élagage des accents, majuscules..., fonctionnent comme attendu. Mais des bogues ont pu ne pas être révélés. Dans ce cas, contactez nous à l'adresse suivante :

[Snowandy5@gmail.com](mailto:Snowandy5@gmail.com)

Nous nous efforcerons de corriger ce problème le plus rapidement possible.

#### **6.2.2 Fonctionnement du groupe de travail**

Le projet a été terminé dans les temps, avec une bonne entente entre nous. Aucun problème majeur n'a été rencontré dans notre chemin.

Ce mini projet a été une expérience enrichissante dans notre apprentissage de la programmation. C'est aussi un avant-goût du projet final que nous allons devoir réaliser pour le bac. Nous en retirons un excellent souvenir, d'un projet ludique, et non d'un devoir imposé.

## 7. Annexe : Listings identifiés et commentés

### Code source :

```
1  import unicodedata #librairie utilisé pour enlever les accents
2  de la chaîne
3
4
5  def palindrome(): #on crée la fonction qui détermine la
6  validité du palindrome
7
8      for i in range(0,len(phrase)):
9          if phrase[i] == phrase[len(phrase)-1-i]: #on test la
10 i valeur avec celle qui correspond(la dernière valeur-la valeur de
11 i)
12             print("etape",i,"correcte")
13             resultat=1
14         else:
15             resultat=2
16             print("etape",i,"incorrecte")
17             break #arrêt de la boucle. ce n'est pas un
18 palindrome
19
20     if resultat==1:
21         print("votre phrase est un palindrome")
22     else:
23         print("votre phrase n'est pas un palindrome")
24     return
25
26
27
28 #debut du programme
29 print("le programme devra distinguer si c'est un palindrome en
30 tenant compte des accents et majuscules, tapez 1")
31 print("sinon entrez 2") #interface menu.
32
33 recommencer='o' #initialisation de la variable recommencer
34
35 while recommencer == 'o': #boucle qui permet a la fin, de
36 choisir de recommencer ou non le programme
37     print("votre choix? (choix 1 par défaut)")
38     choix=str(input())
39     print("entrez maintenant votre phrase")
40     phrase = input()
41
42
43     phrase = phrase.strip() #élimine les espaces parasite au
44 début et à la fin
45     phrase = phrase.replace(" ", "") #supprime tous les espaces
46
47
48     if choix== '2':
49         print("choix 2")
50
51     #simplification de la phrase au maximum
```

*la chasse aux palindromes*

```
62
63     phrase = phrase.lower() #enleve les majuscules
64     phrase =
65     unicodedata.normalize('NFKD',phrase).encode('ASCII', 'ignore')
66     #ligne qui permet de convertir de l'unicode en
67     ASCII(destruction des accents)
68
69     palindrome() #on appelle la fonction de détermination
70     du palindrome
71
72     else:
73         print("choix 1")
74         palindrome()
75
76         print("voulez vous recommencer? o/n")
77         recommencer=input() #choix pour recommencer la boucle, donc le
78     programme
```