Mini-projets :

HALTE ! Mot de passe !

Lycée Charles Gide

Rapport de projet informatique de terminale scientifique,
spécialité ISN
du 25/11/13 au 16/12/13

CARBON Dimitri TS2
CHABALIER Andy TS2

pour profiter pleinement de la mise en page du code, imprimez en couleur les dernière pages (la couleur syntaxique est respectée)

Table des Matières

1. Introduction

1.1 Généralités

Le mini-projet de programmation nous permet de nous entraîner a la réalisation d'un programme, ainsi que d'un compte rendu de projet. D'une durée de 21 jours, l'exercice doit nous permettre de nous aider a comprendre les exigences du bac. Encadrer par notre Professeur, chaque couple d'élève doit remettre un programme fonctionnel et un rapport de projet.

1.2 Le sujet

Pour sécuriser des informations sur internet, il est demandé aux utilisateurs de saisir un mot de passe.

L'objectif de ce mini projet est d'écrire un programme sous Python qui vérifie le niveau de sécurité de ce mot de passe et s'il n'y a pas eu d'erreur de saisie.

Les vérifications pourront se faire soit à l'issue de la 2nde saisie, soit au fur et à mesure.

Vous inviterez l'utilisateur à saisir un nouveau mot de passe dans les cas suivants :

– mot de passe non identique
– mot de passe trop court (4 caractères ou moins)
- mot de passe trop long (plus de 10 caractères)
– mot de passe non conforme (2 chiffres ou moins)

Vous rédigerez un rapport dans lequel figurera le code Python et la stratégie adoptée pour répondre à chaque exigence du sujet.

1.3 Cahier des charges

Le programme doit pouvoir déterminer si un mot de passe appartient a une de ces catégories, et dans ce cas la, inviter l'utilisateur a saisir un autre mot de passe :

– mot de passe non identique
– mot de passe trop court (4 caractères ou moins)
- mot de passe trop long (plus de 10 caractères)
– mot de passe non conforme (2 chiffres ou moins)

le programme doit donc vérifier si le mot de passe saisis contient entre 4 et 10 caractères, dont au moins 2 chiffres

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 aboli, 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 intégrer. Nous avons choisis 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

Pour mettre en œuvre le projet, nous nous somme penché dans un premier temps, sur une fonction qui permettrait de demander a l'utilisateur de saisir le mot de passe, puis de le vérifier.

Pour cela nous sommes partit sur une boucle tan que, qui tournerai jusqu'à que le mot de passe saisis sois le même que celui saisit dans la vérification.

Puis nous nous sommes penché sur le moyen de vérifier la plongeur du mot de passe.

Nous avons choisis de crée une double condition :

-si le mot de passe a moins de 4 caractères,

-si le mot de passe a plus de 10 caractères

dans ces deux cas, la fonction de vérification doit se réenclenché, pour demander un autre mot de passe.

Si le mot de passe est de bonne longueur, la dernière vérification s'enclenche.

Pour savoir si le mot de passe contient au minimum 3 chiffres, nous avons décider de tester si chaque caractère du mot de passe, est un des 10 chiffres, a l'aide d'une boucle for.

4. Analyse technique

```
1   def demandmdp(mot,ver):
2       while mot!=ver:
3            mot=input("mauvais mot de passe: entrez a nouveau votre mot
4 de passe: ")
5           ver=input("verifiez votre mot de passe: ")
6
7   def typecarac(mot):
8       nbent=0
9       for k in range(0,len(mot)):
10                if mot[k]== '0' or mot[k]== '1' or mot[k]== '2' or
11 mot[k]=='3'  or  mot[k]=='4'  or  mot[k]=='5'  or  mot[k]=='6'  or
12 mot[k]=='7' or mot[k]=='8' or mot[k]=='9':
13                nbent=nbent+1
14            else:
15                print ()
16
17      if nbent>2:
18          print("mot de passe conforme")
19      else:
20           print("mot de passe non conforme, il doit contenir au moins
21 3 chiffres")
```

On retrouve ici les deux fonctions essentielles du programme.

La première demande a l'utilisateur de saisir son mot de passe puis de le vérifier. Tant que le mot de passe est différent de la vérification, l'utilisateur doit saisir son mot de passe.

La seconde permet de connaître le nombre de chiffres présent dans le mot de passe.
Nbent est la variable qui s'incrémentera de une unité pour chaque chiffres du mot de passe.
La boucle for permet au programme de balayer chaque caractère a la recherche des chiffres, grâce a la condition if mot[k]==..... (ce qui en sois n'est pas extrêmement propre.)

puis on teste nbent avec la plafond minimum de chiffre qui doivent être présent dans le mot de passe.

```
35   if len(motdepasse)<4:
36           print("mot de passe trop court")
37           demandmdp(motdepasse,vermotdepasse)
38
39       elif len(motdepasse)>10:
40           print("mot de passe trop long")
41           demandmdp(motdepasse,vermotdepasse)
42       else:
43           print("mot de passe de bonne longueur")
44
45
46       typecarac(motdepasse)
```

Ici, on vérifie la longueur du mot de passe. Si la taille du mot de passe est mauvaise, on appelle la fonction pour redemander un mot de passe conforme.

Puis on appelle la dernière fonction pour connaître le niveau de sécurité

5. Manuel d’utilisation

Après avoir exécuter le programme, l'utilisateur devra saisir son mot de passe, puis le vérifier. Le programme déterminera seul si le mot de passe est sur, en deux étapes :

-si le mot de passe est de bonne longueur, on passe a l’étape suivante, sinon, l'utilisateur devra saisir et vérifier un autre mot de passe.

-la seconde étape vérifie le niveau de sécurité 2, c'est a dire si le mot de passe comporte au moins 3 chiffres.

6. Perspectives et conclusions

6.1 Perspectives

Ce programme peut être modifier et améliorer énormément. Tout d'abord, il peut servir en l'état à connaître si son mot de passe est de niveau 2, et correspond aux canons d'inscription du logiciel (ou du site). Mais il peut être améliorer en augmentant le champs de vérification, le choix du niveau de protection. On pourrait même proposer de générer automatiquement des mots de passe. Pour résumer, le champs d'amélioration de ce type de programme est limité par notre imagination.

6.2 Conclusions

6.2.1 Fonctionnement de l’application

Le programme bien que relativement simple, fonctionne comme il était demander. Le seul bémol, est sa sur-simplicité. En effet, l'utilisateur n'a pas énormément de choix, bien que se sois modifiable.

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 ont été dans notre chemin. Malgré cela, le manque de temps dut à d'autres travaux dans d'autre matière nous a limité dans l’expansion du programme, nous avons dut nous limiter aux sujet.

Mais 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 :

```
   def demandmdp(mot,ver):
       while mot!=ver:
            mot=input("mauvais mot de passe: entrez a nouveau votre mot
de passe: ")
           ver=input("verifiez votre mot de passe: ")

   def typecarac(mot):
       nbent=0
       for k in range(0,len(mot)):
                if mot[k]== '0' or mot[k]== '1' or mot[k]== '2' or
mot[k]=='3' or mot[k]=='4' or mot[k]=='5' or mot[k]=='6' or mot[k]=='7'
or mot[k]=='8' or mot[k]=='9':
                nbent=nbent+1
           else:
               print ()

       if nbent>2:
           print("mot de passe conforme")
       else:
           print("mot de passe non conforme, il doit contenir au moins 3
chiffres")

   recommencer='o'
   while recommencer=='o':
       motdepasse=input("entrez votre mot de passe: ")
       vermotdepasse=input("verifiez votre mot de passe: ")

       demandmdp(motdepasse,vermotdepasse)

       print("mot de passe identique")
       print()



       if len(motdepasse)<4:
           print("mot de passe trop court")
           demandmdp(motdepasse,vermotdepasse)

       elif len(motdepasse)>10:
           print("mot de passe trop long")
           demandmdp(motdepasse,vermotdepasse)
       else:
           print("mot de passe de bonne longueur")


       typecarac(motdepasse)


       print("voulez vous recommencer? o/n")
        recommencer=input()   #choix pour recommencer la boucle, donc le
programme
```