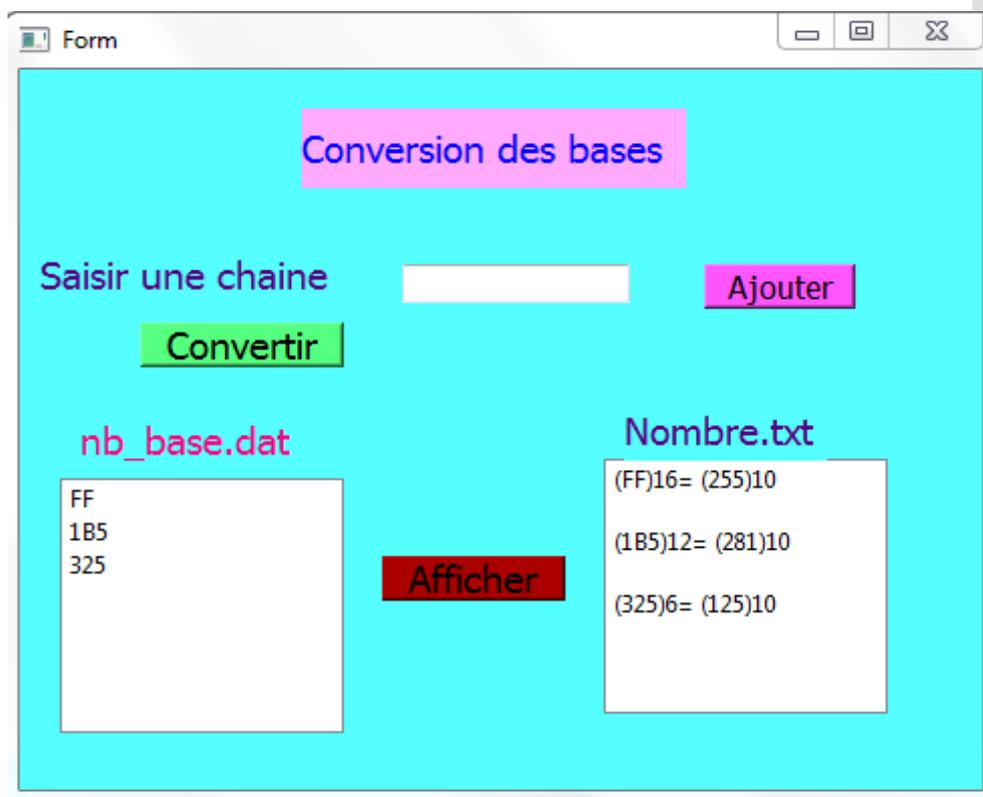


Application 5(Bac TP) Avec Qt designer

Soit un fichier enregistré sous le nom "**nb_base.dat**" dans lequel on va enregistrer n chaînes de caractères. Chaque chaîne est formée de 5 caractères au maximum. Chacun de ces caractères est soit un chiffre, soit une lettre majuscule de l'intervalle "A".."F".

Pour chaque chaîne du fichier "**nb_base.dat**", on se propose de réaliser les tâches suivantes :

- Déterminer la plus petite base **b** à laquelle, peut appartenir le nombre **m** représenté par la chaîne de caractères.
- Déterminer l'équivalent décimal **d** du nombre **m**.
- Ranger dans une ligne d'un fichier texte nommé "**nombre.txt**". la valeur décimale correspondante, de la façon suivante : $(m) b = (d) 10$.



Remarque :

La plus petite base à laquelle appartient la valeur **1B5** est la base **12**. En effet, le plus grand chiffre de ce nombre est B qui correspond à la valeur décimale **11**.

De ce fait, la plus petite base de **1B5** est $11 + 1 = 12$

Travail demandé :

Ecrire un programme qui permet de :

- Enregistrer dans un fichier nommé **nb_base.dat**. des chaînes de caractères répondant aux conditions précédemment citées en utilisant le bouton **Ajouter**.
- Remplir le fichier **nombre.txt**
- Afficher les fichiers **nombre.txt** et **nb_base.dat**.

Correction

```
from PyQt5.uic import loadUi
from PyQt5.QtWidgets import QApplication, QMessageBox, QTableWidgetItem
from pickle import load, dump
```

```
def valide(ch):
```

```
    B=True
    i=0
    while( B and i<len(ch)):
        if ("A"<=ch[i]<="F" or
"0"<=ch[i]<="9"):
            i=i+1
        else:
            B=False
    return B
```

```
def converb_10(ch):
```

```
    b=maxi(ch)
    s = 0
    p = 1
    for i in range(len(ch)-1,-1,-1):
        if "0"<=ch[i]<="9":
            c=int(ch[i])
        else:
            c=ord(ch[i]) -55
        s=s+ c* p
        p = p * b
```

```
def convertir():
```

```
    fb=open("base.dat", "rb")
    fc=open("nombr.txt", "w")
    fin=False
    while fin==False:
        try:
            ch=load(fb)
            y=converb_10(ch)
            fc.write((" "+ch+"")+str(maxi(ch))+ " (" +str(y)+")10\n")
        except:
            fin=True
    fb.close()
```

```
def afficher():
```

```
    fb=open("base.dat", "rb")
    fin=False
    while fin==False:
        try:
            ch=load(fb)
            w.f1.addItem(str(ch))
        except:
            fin=True
    fb.close()
    fc=open("nombr.txt", "r")
    ch1=fc.readline()
```

```
def maxi(ch):
```

```
    m=ch[0]
    for i in range(len(ch)):
        if ch[i]>m:
            m=ch[i]
    if "0"<=m<="9":
        return int(m)+1
    else:
        return (ord(m)-55)+1
```

```
    return s
```

```
def ajouter():
```

```
    fb=open("base.dat", "ab")
    ch=w.t1.text()
    if valide(ch)==False:
```

```
        QMessageBox.critical(w, "attention", "verifié")
```

```
    else:
        dump(ch,fb)
        w.t1.clear()
```

```
w.f2.addItem(ch1)
while ch1!="":
    ch1=fc.readline()
    w.f2.addItem(ch1)
```

```
app = QApplication([])
w = loadUi("app2.ui")
w.show()
fb=open("base.dat","wb")
w.b1.clicked.connect(ajouter)
fb=open("base.dat","rb")
w.b3.clicked.connect(afficher)
w.b2.clicked.connect(convertir)
app.exec_()
```

JALEL SAADI