Corrigé devoir surveillé 1

Exercice 1 (5 pts)

1. L'appel `d(4)` renvoie la liste `[1,2,4]`. L'appel `d(10)` renvoie la liste `[1,2,5,10]`. `d(n)` renvoie la liste des diviseurs de `n`.
2.

```
def DNT(n):
    L=[nombre for nombre in range(2,n) if n%nombre==0]
    return L
```

3.

```
def sommeCarresDNT(n):
    s=0
    for u in DNT(n):
        s+=u**2
    return s
```

4.

```
for i in range(2,1001):
    if i==sommeCarresDNT(i):
        print(i)
```

Exercice 2 (3 pts)

1.

```
def mini(liste):
    nblettres=len(liste[0])
    ind=0
    for i in range(len(liste)):
        if len(liste[i])<nblettres:
            nblettres=len(liste[i])
            ind=i
    return liste[ind],ind
```

2. Pour le deuxième cas, il suffit de remplacer dans le test l'inégalité stricte par une inégalité large : `len(liste[i])<=nblettres`.

Exercice 3 (4 pts)

1.

```
from math import exp,sin
def g(x):
    return exp(sin(x))
```

2.

```
import matplotlib.pyplot as plt
def trace(f,a,b,n):
    pas=(b-a)/(n-1)
    x=[a+i*pas for i in range(n)]
    y=[f(u) for u in x]
    plt.plot(x,y)
    plt.show()

from math import pi
trace(g,-pi,pi,100)
```

Exercice 4 (8 pts)

1.

```
def parentheses(ch):
    par=''
    for u in ch:
        if u=='(' or u==')':
            par=par+u
    return par
```

2.

```
def nombres(m):
    n1,n2=0,0
    for u in m:
        if u=='(':
            n1+=1
        else:
            n2+=1
    return [n1,n2]
```

3.

```
def compare(m):
    a,b=nombres(m)
    return a==b
```

4.

```
def verification(m):
    p=creer_pile(len(m))
    couple=[]
    for i in range(len(m)):
        if m[i]=='(':
            empiler(p,i)
        else:
            if pile_vide(p):
                return False
            else:
                j=depiler(p)
                couple.append([j,i])
    if not pile_vide(p):
        return False
    return couple
```

5. L'appel `parentheses("()(()())")` renvoie `"()(()())"`.

Il aurait été bien sûr plus judicieux de demander l'appel `verification("()(()())")` qui renvoie la liste `[[0,1],[3,4],[5,6],[2,7]]`.