Fonctions

1. Définir une fonction **f(x)** qui correspond à la fonction mathématique $f : x \mapsto x^3 - 2x + 1$. Faire afficher la valeur $f(1) + f(2)$.

Réponse : _____

2. Au début du programme, la commande

```
from math import cos, sin, exp, log
```

permet d'avoir accès aux fonctions cos, sin, exp, ln. Définir une fonction **g(x)** qui correspond à la fonction mathématique

$$g : x \mapsto \frac{\cos x + 1}{2(\mathrm{e}^x + 1)}$$

Faire afficher la valeur $2 \times g(3)$.

Réponse : _____

3. Écrire une fonction **S(x,n)** qui calcule et renvoie la valeur de la somme :

$$S_n(x) = \sum_{k=1}^{n} \sin(kx)$$

Faire afficher $S_5(0)$, $S_5(1)$ et $S_7(2)^2$.

Réponse : _____

4. Écrire une fonction **minimum(a,b)** qui renvoie le plus petit des deux nombres a et b. Tester la fonction avec **minimum(1,1)**, **minimum(1,2)** et **minimum(2,-1)**.

5. On considère la suite (u_n) définie par :

$$u_0 = 1 \quad \text{et} \quad \forall n \in \mathbb{N},\ u_{n+1} = 2u_n^2 + 1$$

Écrire une fonction **u(n)** qui calcule et renvoie la valeur de u_n. Faire afficher u_5.

Réponse : _____

6. Écrire une fonction `Lu(n)` qui construit et renvoie la liste $[u_0, u_1, \ldots, u_n]$. Tester avec `Lu(5)`.

Réponse : _____

7. On considère la suite (v_n) définie par :

$$v_0 = 1 \quad \text{et} \quad \forall n \in \mathbb{N},\ v_{n+1} = 2v_n^2 + n$$

Écrire une fonction `v(n)` qui calcule et renvoie la valeur de v_n. Faire afficher v_5.

Réponse : _____

8. Écrire une fonction `vv(M)` qui renvoie la première valeur de la suite (v_n) pour laquelle $v_n \geqslant M$. Faire afficher `vv(50)`.

9. Écrire une fonction `S(f,n)` qui calcule et renvoie la somme :

$$f(1) + f(2) + \cdots + f(n)$$

(le paramètre `f` est donc lui même une fonction). Faire afficher `S(cos,5)`.

Réponse : _____

Représentations graphiques

10. Définir la fonction `f(x)` qui calcule et renvoie $\cos(x)\exp(-x)$.

11. Construire une liste `X` contenant les valeurs 0, 0.1, 0.2, 0.3 etc. jusqu'à 10.

12. Construire la liste `Y` contenant les valeurs $f(x)$ pour chaque x de la liste `X`.

13. Obtenir la représentation graphique avec les commandes :

```
import matplotlib.pyplot as plt
plt.plot(X,Y,'-')
plt.show()
```

Corrections

Q 1.

```
def f(x):
    return x**3-2*x+1

print(f(1)+f(2))
```

```
5
```

Q 2.

```
from math import cos, sin, exp, log

def g(x):
    return (cos(x)+1)/2/(exp(x)+1)

print(2*g(3))
```

```
0.00047461458705134416
```

Q 3.

```
def S(x,n):
    s = 0
    for k in range(1,n+1):
        s = s+sin(k*x)
    return s

print(S(5,0))
print(S(5,1))
print(S(7,2)**2)
```

```
0
-0.9589242746631385
2.714565838620439
```

Q 5.

```
def u(n):
    U = 1
    for k in range(0,n):
        U = 2*U**2+1
    return U

print(u(5))
```

```
2185969041363
```

Q 6.

```
def Lu(n):
    U = 1
    L = [U]
    for k in range(0,n):
        U = 2*U**2+1
        L.append(U)
    return L

print(Lu(5))
```

```
[1, 3, 19, 723, 1045459, 2185969041363]
```

Q 7.

```
def v(n):
    V = 1
    for k in range(0,n):
        V = 2*V**2+k
    return V

print(v(5))
```

```
5787804054
```

Q 8.

```
def vv(M):
    V = 1
    k = 0
    while V<M:
        V = 2*V**2+k
        k = k+1
    return V

print(vv(50))
```

```
164
```

Q 9.

```
def S(f,n):
    s = 0
    for k in range(1,n+1):
        s = s+f(k)
    return s

print(S(cos,5))
```

```
-1.2358184626798336
```

Q 10.

```
def f(x):
    return cos(x)*exp(-x)
```

Q 11.

```
X = []
for k in range(0,101):
    X.append(0.1*k)
```

Q 12.

```
Y = []
for x in X:
    Y.append(f(x))
```

Q 13.

```
import matplotlib.pyplot as plt
plt.plot(X,Y,'-')
```