NOM :
PRENOM :

Bonus/malus de cahier de colle :

Question 1 (/2 pts). Écrire une fonction prenant en argument n et renvoyant $\prod_{k=2}^{n} \frac{1}{k^2-1}$

```
def prod(n):
    p = 1
    for k in range(2,n+1):
        p = p * 1/(k**2-1)
    return p
```

Question 2 (/3 pts). Soit (u_n) la suite définie par $u_1 = 3$ et : $\forall n \geq 1,\ u_{n+1} = \frac{n+u_n}{1+u_n}$. Écrire une fonction prenant en argument n et renvoyant u_n.

```
def suite(n):
    u = 3
    for k in range(1,n):
        u = (k+u)/(1+u)
    return u
```

Question 3 (/5 pts). Soit (u_n) la suite définie par $u_0 = 2,\ u_1 = 1$ et : $\forall n \in \mathbb{N},\ u_{n+2} = \frac{u_{n+1}}{u_n}$.

1. Écrire une fonction `suite` prenant en argument n et renvoyant u_n.
2. M. Ducallia conjecture que : $\forall n \in \mathbb{N},\ u_n = 2^{1-n}$. En utilisant la fonction `suite`, écrire une fonction `test` prenant en argument n et renvoyant `True` si $u_n = 2^{1-n}$ et `False` sinon. *On prendra soin d'écrire la fonction de la manière la plus simple possible.*

1)
```
def suite(n):
    u,v = 2,1
    for k in range(2,n+1):
        u,v = v, v/u
    return v
```

2)
```
def test(n):
    u = suite(n)
    x = 2**(1-n)
    return u==x
```

NOM :
PRENOM :

Bonus/malus de cahier de colle :

Question 1 (/2 pts). Écrire une fonction prenant en argument n et renvoyant $\prod_{k=2}^{n} \sqrt{k+1}$

```
def prod(n):
    p = 1
    for k in range(2,n+1):
        p = p*(k+1)**(1/2)
    return p
```

Question 2 (/3 pts). Soit (u_n) la suite définie par $u_1 = 3$ et : $\forall n \geq 1,\ u_{n+1} = \dfrac{2+u_n}{n+u_n}$. Écrire une fonction prenant en argument n et renvoyant u_n.

```
def suite(n):
    u = 3
    for k in range(1,n):
        u = (2+u)/(k+u)
    return u
```

Question 3 (/5 pts). Soit (u_n) la suite définie par $u_0 = \frac{1}{3}$, $u_1 = 1$ et : $\forall n \in \mathbb{N},\ u_{n+2} = \dfrac{u_n}{u_{n+1}}$.

1. Écrire une fonction `suite` prenant en argument n et renvoyant u_n.
2. M. Ducallia conjecture que : $\forall n \in \mathbb{N},\ u_n = 3^{n-1}$. En utilisant la fonction `suite`, écrire une fonction `test` prenant en argument n et renvoyant `True` si $u_n = 3^{n-1}$ et `False` sinon. *On prendra soin d'écrire la fonction de la manière la plus simple possible.*

1)
```
def suite(n):
    u,v = 1/3, 1
    for k in range(2,n+1):
        u,v = v, u/v
    return v
```

2)
```
def test(n):
    u = suite(n)
    x = 3**(n-1)
    return u==x
```