Prédiction du cycle menstruel

Sujet n°3 — Corrigé

Question 1 *Années bissextiles*

L'ordre des tests compte : 400 avant 100, et 100 avant 4.

```
def est_bissextile(annee):
    if annee % 400 == 0:
        return True
    if annee % 100 == 0:
        return False
    return annee % 4 == 0
```

2024 et 2000 sont bissextiles, 2100 et 2025 ne le sont pas.

Question 2 *Phase d'un jour du cycle*

```
def determiner_phase(jour):
    assert 1 <= jour <= 28, "le jour du cycle doit être compris entre 1 et 28"
    if jour <= 5:
        return 1
    elif jour <= 13:
        return 2
    elif jour == 14:
        return 3
    else:
        return 4
```

Question 3 *Tests de `ajouter_jours`*

```
def test_ajouter_jours():
    assert ajouter_jours((7, 9, 2025), 3) == (10, 9, 2025)
    # passage au mois suivant : 30 jours en septembre
    assert ajouter_jours((28, 9, 2025), 5) == (3, 10, 2025)
    # passage à l'année suivante
    assert ajouter_jours((30, 12, 2025), 3) == (2, 1, 2026)
    # février d'une année bissextile : le 29 existe
    assert ajouter_jours((28, 2, 2024), 1) == (29, 2, 2024)
    # février d'une année non bissextile : on passe au 1er mars
    assert ajouter_jours((28, 2, 2025), 1) == (1, 3, 2025)
    # ajout de 0 jour : la date est inchangée
    assert ajouter_jours((15, 6, 2026), 0) == (15, 6, 2026)
```

Question 4 *Un calendrier invalide*

Cause : `str(annee) + str(mois) + str(jour)` n'ajoute pas les zéros de tête. Le 12 mars 2026 donne `DTSTART:2026312` (7 caractères au lieu de 8), date que le module `ics` ne sait pas relire. **Correction** : compléter mois et jour sur deux chiffres.

```
        date = str(annee) + str(mois).zfill(2) + str(jour).zfill(2)
```

À partir du 12 mars 2026, le calendrier contient `20260312`, `20260409` et `20260507`, et `ics` le lit sans erreur.

Remarque : la condition `jours_ecoules + 28 <= 100` écarte la date du 4 juin 2026 (jour 84), pourtant située dans les 100 jours ; la condition `jours_ecoules <= 100` la retiendrait.