Exercice 1 :

Écrire un programme qui lit deux entiers et renvoie la valeur absolue de ceux-ci.

Exercice 2 :

Écrire un programme qui demande la saisie d'un nombre N et renvoie N étoiles. Par exemple si $N = 3$, le programme renvoie

```
*
*
*
```

Exercice 3 :

Écrire un programme qui demande la saisie d'un nombre N et renvoie N étoiles. Par exemple si $N = 3$, le programme renvoie

```
***
```

Exercice 4 :

Écrire un programme qui demande la saisie d'un nombre N et renvoie un triangle isocèle rectangle de coté N étoiles. Par exemple si $N = 3$, le programme renvoie :

```
*
**
***
```

Exercice 5 :

Écrire un programme qui demande la saisie d'un nombre N et renvoie u_N étoiles. Avec u définie par $u_0 = 5$ et $u_{n+1} = 3u_n + 2$

Exercice 6 :

Écrire un programme qui demande la saisie réel $A > 2$ et renvoie le plus petit entier n tel que $u_n > A$ avec u définie par $u_0 = 2$ et $u_{n+1} = u_n + 2$

Exercice 7 :

Récupérer sur la page web et enregistrer dans le dossier personnel les fichiers image.py et dumas.jpg.

(1) Ouvrir le fichier image.py et analyser le programme.

(2) Modifier l'image par une symétrie centrale par rapport au centre de l'image.

(3) On joue maintenant sur la luminosité de l'image : puisque les valeurs les plus hautes sont les plus claires, il suffit d'augmenter de 50 les valeurs des pixels pour éclaircir l'image et de les diminuer pour l'assombrir.
On s'aperçoit vite que l'effet obtenu n'est pas celui escompté : tous les pixels dont la valeur était supérieure à 205 reçoivent une valeur entre 255 et 305. Comme les valeurs des pixels sont comptées modulo 255, ces pixels deviennent en fait presque noirs.
Écrire une fonction qui éclaircit une image en s'assurant par un test que la valeur des pixels est plafonnée à 255.
Est-ce satisfaisant ?

(4) Effectuer une symétrie axiale de l'image (on remplacera chaque image_tab[i][j] par image_tab[-i][j])

(5) Effectuer une symétrie centrale de l'image.

(6) Effectuer une rotation de 90 degré de cette image.