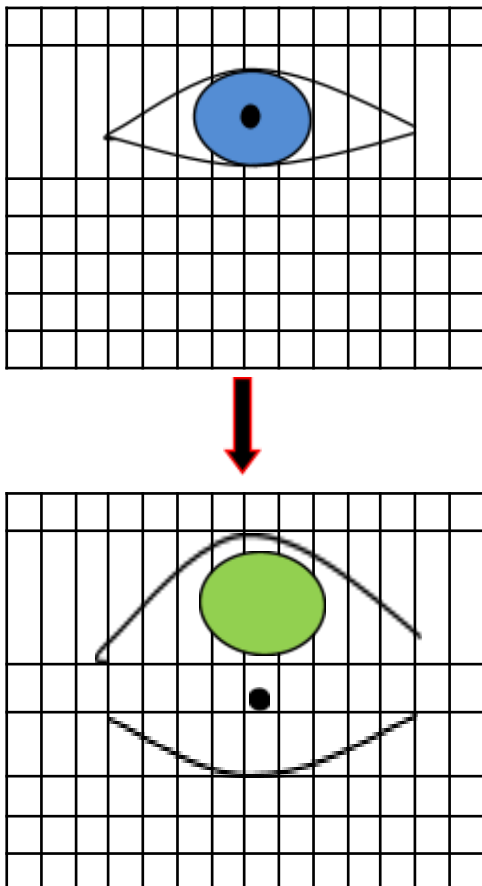


Exercice 1

Une image en informatique peut être représentée par une matrice de points ayant N lignes et M colonnes. Un élément $\text{Image}[x, y]$ de la matrice représente la couleur du point de coordonnées (x, y) . En utilisant la pile comme structure de donnée, on propose d'écrire une action paramétrée qui, à partir d'un point p donné, étale une couleur c autour de ce point. Pour effectuer le remplissage, on doit aller dans toutes les directions à partir d'un point de départ p . La progression de la couleur étalée s'arrête quand elle rencontre une couleur autre que celle du point p . Les figures suivantes illustrent ce concept, en considérant le point noir se trouvant dans la case $(4, 6)$.



Fonction sommet_pile(E/ p : pile) : cordo
Début

Fin ;

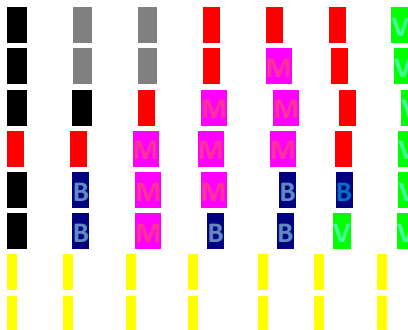
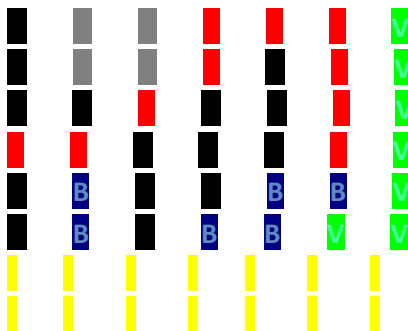
Faire

```

Déempiler (p, z) ;
si (image [z.x][z.y] = c1)
  Alors
    image [z.x][z.y] c ; // mettre à jour cette case avec la nouvelle couleur c
    si (z.x > 1) alors cz z ; cz.x cz.x-1 ; empiler (p, cz) ; fsi ; // haut
    si (z.x < N) alors cz z ; cz.x cz.x+1 ; empiler (p, cz) ; fsi ; // bas
    si (z.y > 1) alors cz z ; cz.y cz.y-1 ; empiler (p, cz) ; fsi ; // gauche
    si (z.y < M) alors cz z ; cz.y cz.y+1 ; empiler (p, cz) ; fsi ; // droite

  Fsi ;
Fait
Fin ;

```



Exercice 3 :

Dans une **chaîne amiable**, chaque nouveau nombre est égal à la **Somme des Diviseurs Propres** (SDP) (autres que lui-même) du précédent. La somme du dernier étant égale au premier nombre. Le nombre en question est dit **sociable** (voir l'exemple donné dans le tableau 1).

N	DIVISEURS PROPRES																SDP
12496	1	2	4	8	11	16	22	44	71	88	142	176	284	568	781	1136	14288
	1562	3124	6248														
14288	1	2	4	8	16	19	38	47	76	94	152	188	304	376	752	893	15472
	1786	3572	7144														
15472	1	2	4	8	16	967	1934	3868	7736								14536

14536	1	2	4	8	23	46	79	92	158	184	316	632	1817	3634	7268	14264
14264	1	2	4	8	1783	3566	7132									12496

Tableau 1 : le nombre 12496 est sociable

1. Ecrire une **fonction récursive** qui vérifie si un nombre N est sociable ou non.
2. Dans le programme principal lisez un entier positif N et afficher le message suivant : N est sociale / N n'est pas sociable.

Remarque :

- Il est recommandé de créer une fonction qui retourne la **SDP** d'un entier N positif.
- Si N n'est pas sociable, la SDP de l' i ème élément ne sera jamais égale au premier. On prévoit alors de transmettre un autre paramètre : borne=100 à la fonction. Si le nombre d'éléments créés dépasse ce paramètre, alors N ne vérifie pas la définition d'une chaîne amiable.

Solution

Fonction som_div(E/ N : entier) : entier

Début

i, som : entier;

som 0;

pour (i 1 à $N/2$)

faire

si ($N \bmod i = 0$) **alors** som som+i; **fsi**;

fait ;

retourner (som) ;

Fin ;

Fonction nb_sociable(E/ N , suivN, borne : entier) : booléen

Début

som : entier ;

som som_div (suivN);

Si (som= N) **alors retourner** vrai;

Sinon

si (borne = 0) **alors retourner** faux;

Sinon borne borne-1;

Retourner nb_sociable (N , som, borne);

Fsi;

Fsi ;

Fin ;

Début

borne, x, N : entier ;

x : booléen ;

borne 100 ;

//N=12496 14316 1264460

Lire (N) ;

x nb_sociable (N, N, borne);

Si (x= vrai) **alors** écrire (N, " est un nombre sociable");

sinon écrire (N, " n'est pas un nombre sociable");

Fin.