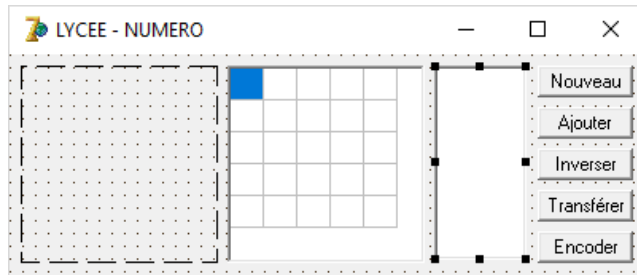




BRANCHE	SECTION(S)	ÉPREUVE ÉCRITE
Informatique partie pratique	B	Durée de l'épreuve : 80 minutes Date de l'épreuve : 07 juin 2018

Créez dans votre dossier de travail l'application Compression (**UMain.pas / Compression.dpr**) qui réalise une compression de type RLE sur une image binaire (noir/blanc). Au lieu d'encoder chaque pixel, une compression RLE encode plutôt le nombre de pixels consécutifs de même couleur. (voir sub 3.)

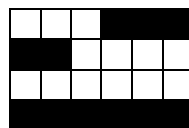


Noms à utiliser :

UMain.pas	Compression.dpr
frmMain	btnNouveau
imgDessin	btnAjouter
sgGrille	btnInverser
lbValeurs	btnTransferer
	btnEncoder

- 1) Reproduisez fidèlement l'interface en vous tenant aux dénominations indiquées. Ajoutez votre LYCEE et NUMÉRO en haut de la fiche et en haut de votre code source. L'image est carrée de taille 153 pixels et toute blanche dès le démarrage du programme. (3+1= 4 p.)
- 2) Réalisez la procédure **Transfert** (`pImage:TImage; var pGrille:TStringGrid`); qui transfère l'image `pImage` pixel par pixel vers la grille `pGrille` en encodant un 'n' pour un pixel noir et un 'b' pour un pixel blanc. Veillez à ajuster la taille de la grille à la taille de l'image (un pixel de l'image correspondant à une case de la grille). (3 p.)
- 3) Réalisez la procédure **Encodage** (`pGrille:TStringGrid; var pListe:TListbox`); qui effectue la compression proprement dite. Pour cela la liste `pListe`, vidée préalablement, contiendra d'abord les dimensions de la grille `pGrille` (largeur, hauteur), puis les séries de pixels consécutifs de même couleur, en commençant par le noir. On parcourt la grille ligne par ligne, de haut en bas, et chaque ligne de gauche à droite. (8 p.)

Exemple pour une image 6x4 pixels :



image

b	b	b	n	n	n
n	n	b	b	b	b
b	b	b	b	b	b
n	n	n	n	n	n

grille

6	largeur
4	hauteur
0	0 pixels noir au début
3	3 pixels blancs
5	5 pixels noirs
10	10 pixels blancs
6	6 pixels noirs

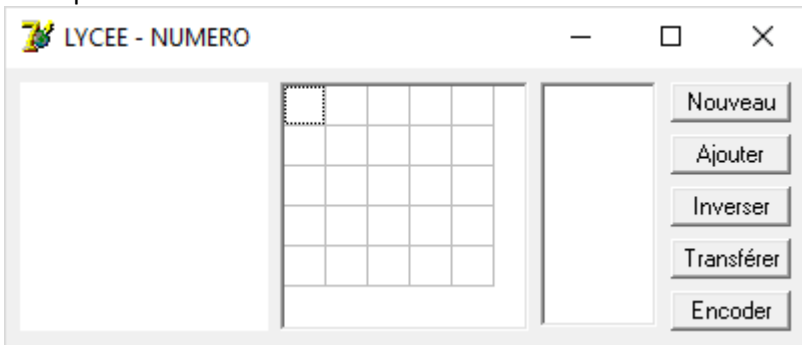
liste

- 4) Un clic sur le bouton `btnNouveau` remet la fiche entière dans l'état de départ (comme au démarrage du programme). (3 p.)
- 5) Si on appuie du bouton gauche de la souris sur `imgDessin`, un disque solide noir (`c1Black`) de diamètre 11 pixels, centré autour des coordonnées de la souris y est ajouté. Si on appuie du bouton droit de la souris sur `imgDessin`, un rectangle solide noir (`c1Black`) de largeur 15 pixels, de hauteur 7 pixels, centré autour des coordonnées de la souris y est ajouté. (3 p.)
- 6) Un clic sur le bouton `btnAjouter` ajoute 6 formes aléatoires comme décrites sous 5) au dessin. Veillez à ce que toutes les formes soient entièrement contenues dans le canevas. (4 p.)

- 7) Un clic sur le bouton **btnInverser** inverse l'image **imgDessin**, donc chaque pixel noir devient blanc et inversement. (3 p.)
- 8) Un clic sur le bouton **btnTransférer** transfère le contenu de **imgDessin** vers **sgGrille** en utilisant la procédure indépendante **Transfert**. (1 p.)
- 9) Un clic sur le bouton **btnEncoder** utilise la procédure indépendante **Encodage** pour compresser les données de **sgGrille** pour les mémoriser dans **lbValeurs**. (1 p.)

Exemples d'exécution :

Au départ :



Après 6 ajouts aléatoires :



Après un transfert et un encodage :

