Nom : ..... Prénom : .....

1B1 Devoir d'informatique II,1 10.03.2017

Partie théorique (25 points - 45 min)

Question 1 — 12 (=8+4) points

(1) Ecrire le code de la fonction **dicho** qui implémente de façon itérative l'algorithme de la recherche dichotomique d'une clé dans une liste de chaînes de type **TListBox**. Préciser la condition que doit remplir la liste pour qu'on puisse utiliser cette fonction.

(2) On donne la liste

```
lbAlphabet :=('a','b','c','d','e',...,'x','y','z')
```

contenant ***toutes les lettres de l'alphabet*** par ordre croissant.

Exécuter en détaillant **dicho(lbAlphabet,'r')** : on donnera dans un tableau d'exécution l'évolution des variables de la fonction et on précisera le résultat retourné.

Question 2 — 13 (=(1+3)+9) points

(1) a) Définir le **type poly** qui permet d'implémenter les polynômes à coefficients réels de degré ≤ 50 dans un programme Delphi.

b) Ecrire la fonction **horner** qui prend en entrée un polynôme **p** de type **poly** et un réel **x** de type **extended** et qui permet de calculer l'image **p(x)** à l'aide de l'algorithme de Horner.

Ecrire la procédure **dessFonction** qui permet de représenter en rouge (sur une image **imgD** de type **TImage** donnée en paramètre) le graphe d'un polynôme donné (non constant) de type **poly**, sur un intervalle donné $[x_1, x_2]$ avec $x_1 < x_2$. L'échelle de l'axe verticale devra s'étendre du ***minimum*** au ***maximum*** du polynôme sur $[x_1, x_2]$. Les images du polynôme seront calculées à l'aide de la fonction **horner** de la question (1). On ne demande pas de représenter les axes du repère.

Partie pratique : Disques tangents et couronnes (35 points - 105 min)

On souhaite créer une application permettant de créer et d'effacer à l'aide de la souris des ***disques tangents intérieurement ou extérieurement*** (1re figure) et des ***couronnes*** (2e figure). Les propriétés des disques à dessiner (***coordonnées en pixels du centre*** = (X,Y), rayon ***entier*** = RAY et couleur (composantes R, G et B) seront inscrits dans le tableau **sgD**.

(1) Créer l'interface graphique **frmMain** ci-dessus avec une image **imgD** et le tableau **sgD** comprenant initialement ***2 lignes*** dont ***une ligne fixe*** et une ligne vide (contrainte technique) ainsi que 6 colonnes, comme sur la figure. ***(2 points)***

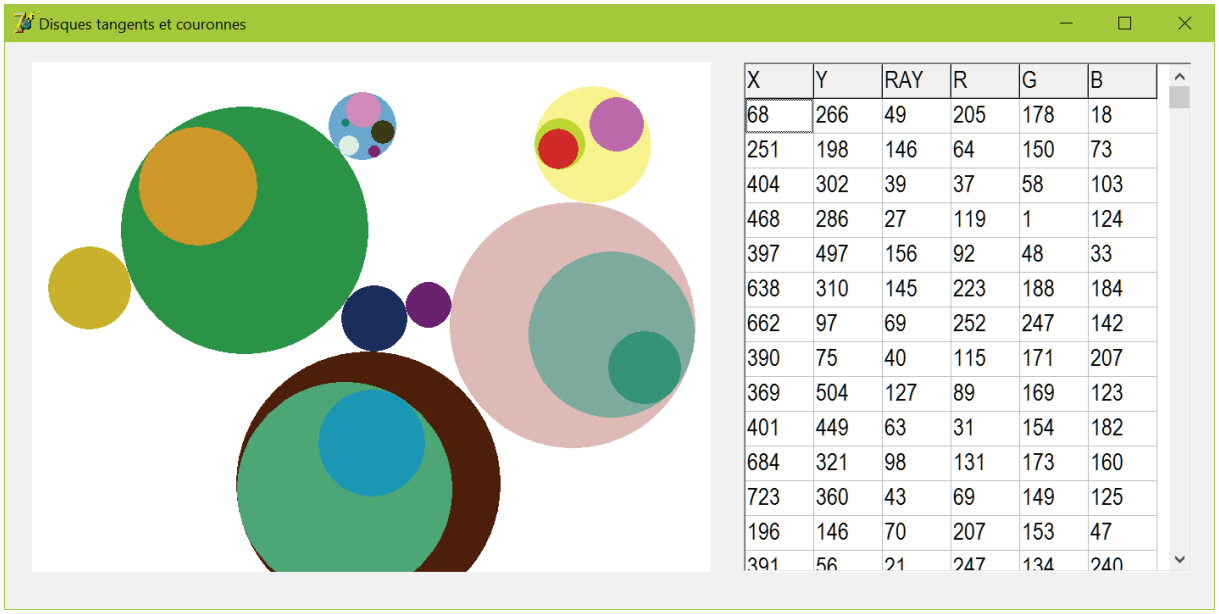


fig. 1

(2) Ecrire la méthode liée à l'événement **OnCreate** du formulaire, permettant d'initialiser le canevas de l'image, le générateur de nombres aléatoires et la ligne fixe du tableau. On initialisera aussi des variables globales éventuelles. ***(2 points)***

(3) Ecrire la fonction **Distance** qui retourne la distance entre deux points **A** et **B** de type **TPoint** donnés en paramètres. (On rappelle qu'une variable **M** de type **TPoint** a deux champs **X** et **Y** de type ***integer***.) ***(2 points)***

(4) Ecrire la procedure **TfrmMain.ActualiserDessin** (sans paramètres) qui efface la figure et qui dessine tous les disques du tableau **sgD**. Attention : le bord d'un disque a même couleur que son intérieur. ***(4 points)***

Tournez s.v.p.

(5) Ecrire la procédure

`TfrmMain.AjouterDisque(O:TPoint;ray,r,g,b:integer)`

qui ajoute d'abord au tableau **`sgD`** le disque de centre **`O`**, de rayon **`ray`** et de couleur RGB de composantes **`r`**, **`g`** et **`b`**, puis redessine tous les disques à l'aide de la procédure **`ActualiserDessin`**. ***(3 points)***

(6) Ecrire la procédure

`TfrmMain.EffacerDisque(numero:integer)`

qui efface d'abord du tableau **`sgD`** le disque de la ligne d'indice **`numero`** (les lignes en-dessous devront être décalées vers le haut) puis redessine tous les disques restants du tableau à l'aide de la procédure **`ActualiserDessin`**. ***(4 points)***

(7) Ecrire la fonction **`TfrmMain.DistMin`** qui pour un point **`M`** de type **`TPoint`** donné en paramètre, retourne ***la plus petite distance*** parmi toutes les distances de **`M`** aux cercles définis dans **`sgD`**. (On rappelle que la ***distance d'un point*** M ***à un cercle*** de centre O et de rayon r est donnée par $\left|OM - r\right|$. Cette formule est valable aussi bien pour un point M situé extérieurement que pour un point M situé intérieurement au cercle.) ***(4 points)***

(8) Ecrire la fonction **`TfrmMain.TrouverDisque`** qui pour un point **`M`** de type **`TPoint`** donné en paramètre, retourne le numéro du disque le plus récent (c.-à-d. le plus en bas dans le tableau) à l'intérieur duquel se trouve le point **`M`**. ***(4 points)***

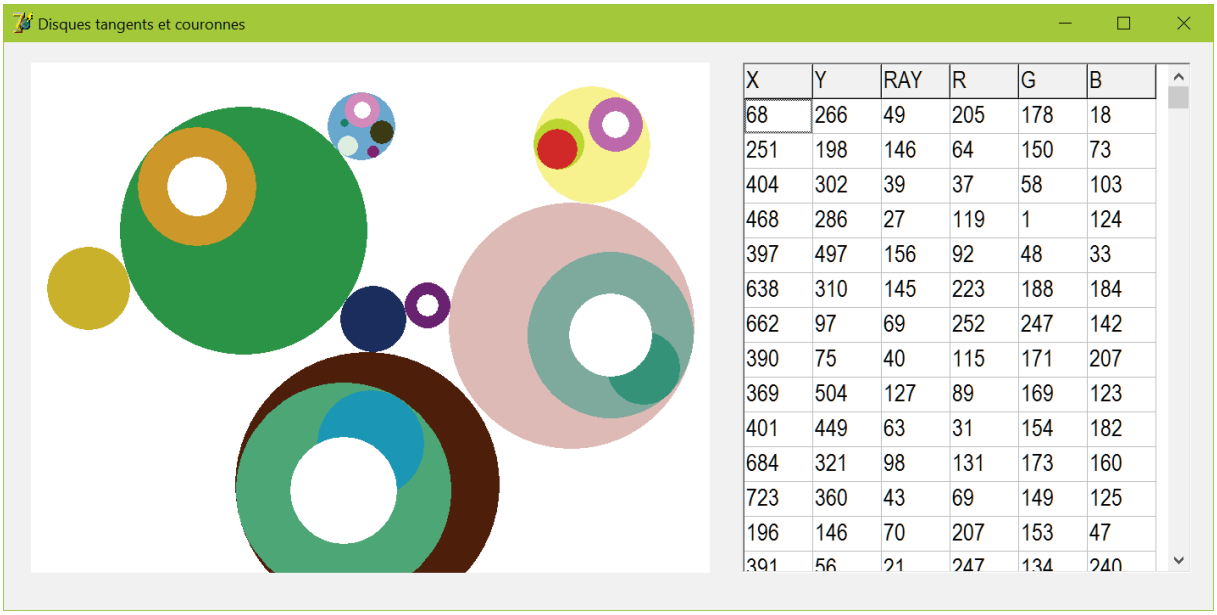
(9) Ecrire la procédure liée à l'événement **`MouseDown`** :

a) Un clic ***avec le bouton gauche*** de la souris ajoute un disque de centre la position de la souris et de couleur aléatoire au tableau (et à la figure). Au premier clic le rayon de ce disque est choisi aléatoirement entre 20 et 50 pixels. Aux clics suivants, le rayon du nouveau disque doit être calculé en sorte que ce disque soit tangent (intérieurement ou extérieurement) au disque le plus proche de la figure. ***Indication*** : utiliser la fonction **`DistMin`**. ***(4 points)***

b) Un clic ***avec le bouton droit*** de la souris efface du tableau (et de la figure) le disque le plus récent à l'intérieur duquel l'utilisateur a cliqué. ***(3 points)***

Tournez s.v.p.

c) Un clic ***avec le bouton gauche et la touche Shift du clavier enfoncé*** permet d'ajouter à l'intérieur du disque le plus récent dans lequel l'utilisateur a cliqué, un disque entièrement blanc, de même centre et de rayon la moitié, donnant ainsi l'illusion qu'on a dessiné une couronne de disque (cf. fig. 2, c'est la fig. 1 avec quelques disques blancs ajoutés). ***(3 points)***



X	Y	RAY	R	G	B
68	266	49	205	178	18
251	198	146	64	150	73
404	302	39	37	58	103
468	286	27	119	1	124
397	497	156	92	48	33
638	310	145	223	188	184
662	97	69	252	247	142
390	75	40	115	171	207
369	504	127	89	169	123
401	449	63	31	154	182
684	321	98	131	173	160
723	360	43	69	149	125
196	146	70	207	153	47
391	56	21	247	134	240

fig. 2