

# Devoir en classe d'Informatique I,2

---

50 minutes – 30 points

## Exercice 1

1.
  - a. Écrivez une fonction qui calcule  $P(a)$  pour un polynôme  $P$  et un nombre  $a$  donnés à l'aide du *Schéma de Horner* (code sans explication). [6 p.]
  - b. Expliquez l'implémentation du polynôme  $P$ . [2 p.]
2.
  - a. Présentez l'algorithme d'une *Puissance à exposant naturel*. [6 p.]
  - b. Modifiez l'algorithme de façon à ce qu'il fonctionne aussi pour les exposants négatifs (code sans explication). [4 p.]

## Exercice 2

L'extrait de code suivant doit afficher le minimum d'une série de notes (entre 1 et 60) contenues dans le composant *lbListe* de type *ListBox* dans le libellé *lblResultat*.

```
min := 0;
for i := 0 to lbListe.Items.Count do
  if lbListe.Items[i] < min then
    min := StrToInt(lbListe.Items[i]);
lblResultat := 'Note minimale : ' + IntToStr(min);
```

1. Recopiez l'extrait de code ci-dessus en éliminant les erreurs logiques et syntaxiques. [6 p.]
2. Illustrez le fonctionnement du code à l'aide d'un tableau d'exécution. Utilisez les nombres suivants comme valeurs des lignes de *lbListe*: 54, 12, 23, 17, 2. [6 p.]

## 70 minutes – 30 points

Selon les règles du jeu d'échecs, une tour peut battre toute autre figure de l'adversaire qui se trouve dans sa rangée ou sa colonne sous condition qu'il ne se trouve pas d'autre figure entre les deux.

Sur l'échiquier suivant, considérons que les figures *a* et *b* appartiennent à l'adversaire, la tour *J* peut battre toutes les figures *a*, mais pas les figures *b*.

				a			
	a			J			a
				a		b	
			b	b			

On donne un échiquier carré avec  $8 \times 8$  cases (représenté par un composant *TStringGrid*) où sont réparties votre tour (lettre J) et les figures de l'adversaire (lettre x). Il s'agit d'écrire une procédure (lancée par un bouton) qui détermine et affiche dans un composant de type *TLabel* le nombre de figures de l'adversaire qui peuvent être battues par votre tour.

S'il n'y a pas de tour ou plusieurs tours, le programme doit afficher un message d'erreur dans le composant de type *TLabel* susmentionné.

1. Interface graphique [8 p.]
2. Écrivez d'abord une procédure *checkrook* (passage par référence) qui détermine l'emplacement de la tour J. (Attention : bonus possible !)  
Si l'échiquier ne contient pas de tour, ou s'il en contient plusieurs, *checkrook* donnera des coordonnées négatives et affichera un message d'erreur. [10 p.]
3. Écrivez ensuite deux fonctions *checkrow* et *checkcol* qui, à partir de l'emplacement de la tour, déterminent combien de pièces peuvent être battues dans la rangée resp. la colonne de la tour. [12 p.]

*Bonus* : Écrivez une fonction *checkrook* (au lieu de la procédure), qui détermine l'emplacement de la tour J. La fonction devra retourner le type *pos*, défini de la manière suivante: [4 p.]

```
type pos = record
  colonne, ligne : integer;
end;
```