Nom : ..... Prénom : .....

1B Devoir d'informatique I,2 12.13.2010

Partie théorique (durée : 45')

Question 1 — 10 (=5+5) points

(1) Ecrire une procédure qui calcule la dérivée d'un polynôme.

(2) Ecrire une procédure qui calcule la primitive s'annulant en 0 d'un polynôme.

Question 2 — 9 (=6+3) points

(1) Ecrire une fonction booléenne **`prime`** qui vérifie si un entier naturel est premier ou non.

(2) En déduire une ***fonction récursive*** **`nextPrime`** qui retourne ***le plus petit nombre premier*** supérieur ou égal à un entier naturel n donné.

Question 3 — 11 (=5+6) points

Soit la procédure ***récursive*** définie par :

```
procedure co(x:integer;lbA:TListBox);
begin
  if x>1 then
    begin
      lbA.Items.Add(inttostr(x));
      if x mod 2 = 0 then co(x div 2,lbA)
                     else co(3*x+1,lbA);
    end
  else lbA.Items.Add('1')
end;
```

(1) Donner le contenu de la liste `lbA` après les instructions suivantes :

```
lbA.Clear ;
co(3,lbA);
```

(2) Ecrire une version ***itérative*** de la procédure `co`.

Partie pratique : Le jeu de taquin (durée : 75')

Le **taquin** est un casse-tête créé vers 1870 aux États-Unis. Il est composé de 15 petits carreaux en bois numérotés de 1 à 15 qui peuvent glisser dans un cadre prévu pour 16 carreaux. Le jeu consiste à remettre dans l'ordre initial (voir fig.1 ; ***case vide en bas à droite***) les 15 carreaux à partir d'une configuration initiale aléatoire (voir fig.2).

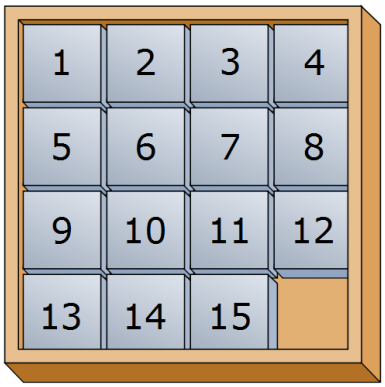

fig. 1: Position initiale

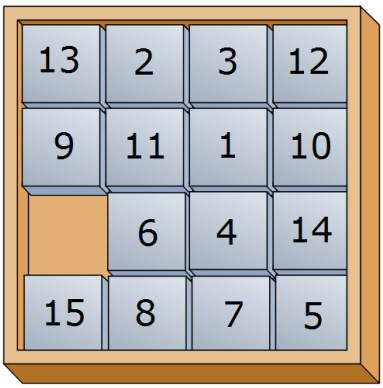

fig. 2 : Taquin mélangé

Le but de la programmation est de créer une simulation du jeu de taquin dans un ***tableau de dimensions quelconques***.

(1) Créer dans Delphi l'interface graphique ci-dessous ***(4 points)*** :

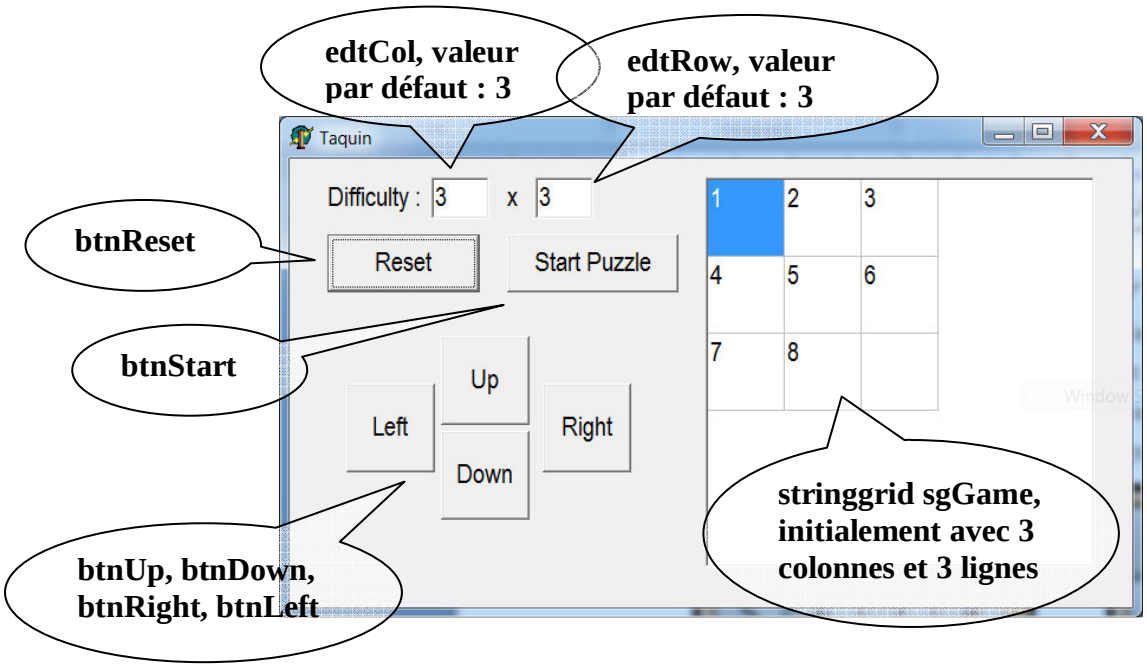

fig. 3 : Interface graphique au démarrage du jeu

(2) Ecrire une procédure indépendante **`gameInit`** qui remplit le tableau **`sgGame`** ligne par ligne avec les nombres de 1 à $c \cdot r - 1$, c étant le nombre de colonnes du tableau, donné dans la boîte d'édition **`edtCol`** et r étant le nombre de lignes du

tableau, donné dans la boîte d'édition **edtRow**. La cellule aux coordonnées (c,r) devra être vide. C'est l'ordre initial du jeu : voir fig. 1 et fig. 3 ***(5 points)***

(3) Ecrire une ***procédure*** indépendante **emptyCell** qui retourne les coordonnées c.-à-d. les indices de la colonne et de la ligne de l'unique cellule vide du tableau **sgGame**. ***(5 points)***

(4) a) Ecrire la procédure indépendante **MoveUp** qui simule le déplacement d'un carreau en bois vers le haut : elle échange donc l'unique case vide du tableau **sgGame** avec la cellule qui se trouve en-dessous de la case vide. Au cas où la cellule vide se trouve dans la dernière ligne du tableau, cette procédure ne devra rien faire ! ***(3 points)*** b) Ecrire de façon analogue les procédures **MoveDown**, **MoveRight** et **MoveLeft**. ***(3 points)***

(5) Au démarrage du programme le générateur de nombres aléatoires devra être initialisé et la procédure **gameInit** devra être appelée pour initialiser le tableau **sgGame**. Ecrire la méthode **FormCreate** correspondante. Un clic sur le bouton **Reset** devra également réinitialiser le tableau **sgGame** à l'aide de la procédure **gameInit**. ***(2 points)***

(6) Ecrire la méthode **btnUpClick** qui devra appeler la procédure **MoveUp**. De façon analogue, un clic sur le bouton **Down** (resp. **Right**, **Left**) devra appeler la procédure **MoveDown** (resp. **MoveRight**, **MoveLeft**). ***(2 points)***

(7) Un clic sur le bouton **Start Puzzle** devra mettre les éléments du tableau dans le désordre en appelant un grand nombre de fois (à choisir en fonction de la rapidité de la machine et en fonction des dimensions du tableau) et ***au hasard*** les procédures **MoveUp**, **MoveDown**, **MoveRight** et **MoveLeft**. ***(6 points)***

(***Attention*** : il est interdit de mettre les éléments dans le désordre par une autre méthode pour la simple raison que toutes les configurations initiales ne permettent pas de résoudre le taquin, c.-à-d. de revenir à l'ordre initial, Par exemple, la configuration initiale choisie par le célèbre Sam Loyd (fig. 4, inversion du 14 et du 15), ne peut pas être résolue !)

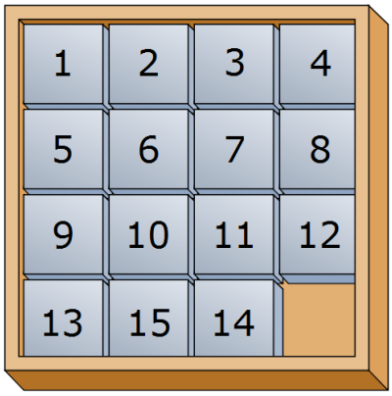

fig. 4 : Configuration de Sam Loyd

G. Lorang