

## Partie théorique (30 points ; durée : 50 min.)

## Question 1

18 (=12+6) points

- (1) Présenter la version itérative du tri croissant par insertion (code complet).
- (2) Illustrer l'algorithme du tri croissant par insertion en l'appliquant au tableau de strings suivant (chaque string est constitué d'un seul caractère) :

'M' 'O' 'T' 'E' 'N'

Ecrire une ligne pour chaque changement dans la liste.

## Question 2

12 (=3+2+4+3) points

Soit la fonction  $f$  définie par :

```
function f(x:string):string;
begin
  if length(x)<=1 then f:=x
  else f:=x[length(x)]+f(copy(x,1,length(x)-1))+x[length(x)]
end;
```

- (1) Quel est le résultat renvoyé par l'appel  $f('MOIEN')$  ?
- (2) Décrire en toute généralité le résultat obtenu par  $f$ , pour tout argument valable  $x$ .
- (3) Ecrire une version itérative de la fonction  $f$ .
- (4) Soit la fonction  $g$  définie par :

```
function g(x:string):string;
var i:integer;
begin
  for i:=2 to length(x) do x:=x[i]+x;
  g:=x
end;
```

Est-ce qu'elle produit toujours le même résultat que la fonction  $f$  ? Discuter en fonction de la longueur du string passé comme argument.

*(Examen de fin d'études secondaires : session de juin 2010)*

## Partie pratique (30 points ; durée : 70 min.)

Le but du problème est d'implémenter dans Delphi le casse-tête mathématique « **M12** ». Ce jeu est inspiré du Rubik's cube et consiste à remettre dans l'ordre les entiers de 1 à 12, permutés aléatoirement, à l'aide de deux opérations élémentaires seulement qui seront expliquées par la suite.

- (1) Créer l'interface graphique avec les éléments suivants (voir *fig. 1*) :
- un tableau (*stringgrid*) **sgJeu** avec 1 ligne et 12 colonnes ;
  - 4 boutons avec les mentions respectives « **Reset** », « **Randomize** », « **Invert** » et « **Merge** » ;
  - 2 libellés (*labels*), dont le premier porte la mention : « **Move History :** » et le deuxième, placé juste derrière le premier et initialement vide (invisible sur la figure ci-dessous), contiendra l'historique des coups joués.

(4 points)

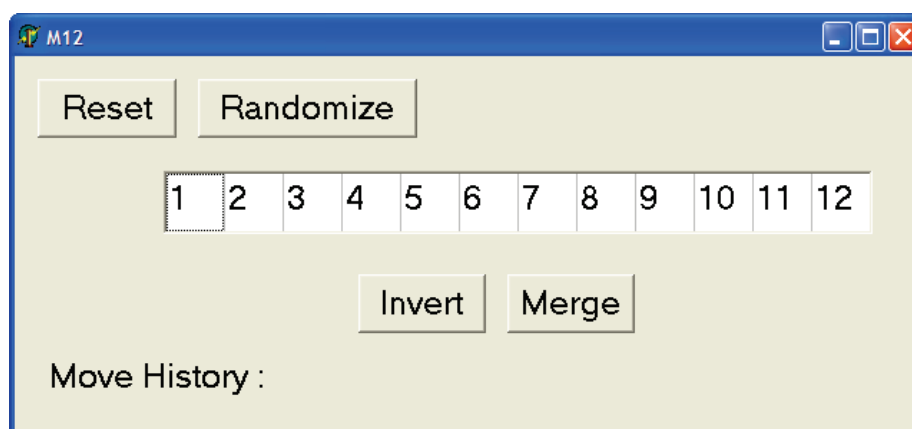


fig. 1 : l'interface du jeu au démarrage, resp. après un clic sur « **Reset** »

- (2) a) Ecrire la méthode **FormCreate**, lancée au démarrage du jeu, qui placera les entiers de 1 à 12 dans l'ordre croissant dans le tableau **sgJeu** et initialisera le générateur de nombres aléatoire (voir *fig. 1*). b) Lorsque le joueur clique sur le bouton « **Reset** », l'historique des coups de la partie précédente sera effacé, et les entiers de 1 à 12 seront remis dans l'ordre dans le tableau **sgJeu** (voir *fig. 1*).

(5 points)

- (3) Pour commencer, le joueur cliquera sur le bouton « **Randomize** », ce qui aura comme effet de mettre les 12 entiers dans un ordre aléatoire dans le tableau **sgJeu** (voir *fig. 2*).

(5 points)

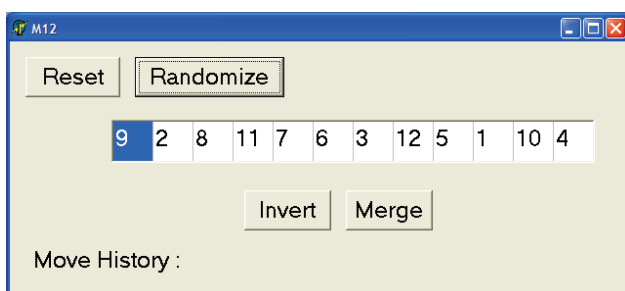
Dans la suite du jeu, le joueur essaiera de remettre les entiers dans l'ordre croissant en utilisant soit le bouton « **Invert** » soit le bouton « **Merge** », autant de fois qu'il le souhaite.

- (4) Un clic sur le bouton « **Invert** » aura comme effet d'inverser l'ordre des entiers dans le tableau. La lettre « **I** » sera ajoutée dans le libellé contenant l'historique (comparer les tableaux des *figures 2* et *3*). Mais : cliquer **deux fois de suite** sur « **Invert** » n'a évidemment pas d'effet sur l'ordre des entiers ; voilà pourquoi l'historique ne devra jamais contenir la séquence « **II** ». En d'autres termes, si le joueur clique pour la 2<sup>e</sup> fois de suite sur « **Invert** », le « **I** » résultant du 1<sup>er</sup> clic devra de nouveau disparaître de l'historique.

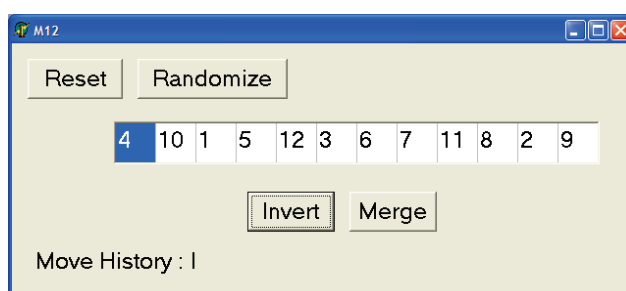
(6 points)

- (5) Un clic sur le bouton « **Merge** » aura comme effet de placer les 6 premiers éléments du tableau **sgJeu** dans les colonnes d'indices pairs sans changer leur ordre et les 6 derniers éléments dans les colonnes d'indices impairs avec inversion de l'ordre (comparer les tableaux des *figures 3 et 4*). La lettre '**M**' sera ajoutée à l'historique. Si le joueur clique 2 (resp. 3, 4, ..., 10) fois de suite sur « **Merge** », la séquence '**M2**' (resp. '**M3**', '**M4**', ... '**M10**') apparaîtra dans l'historique (voir *figures 4 et 5*). Mais : cliquer 11 fois de suite sur le bouton « **Merge** » n'a pas d'effet sur l'ordre des entiers ; dans ce cas la dernière séquence '**M10**' (résultant des 10 premières opérations « **Merge** ») devra disparaître de l'historique. (N.B. La correction tiendra compte de votre style de programmation.)

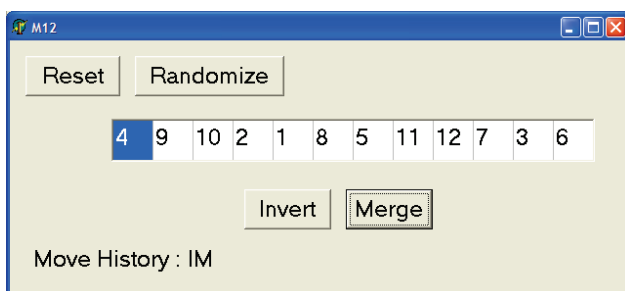
(10 points)



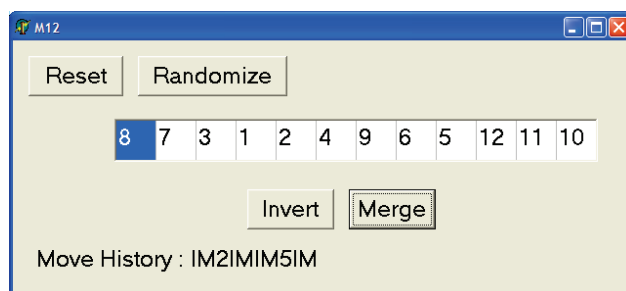
*fig. 2* : après un clic sur « **Randomize** »



*fig 3* : suite du jeu après un clic sur « **Invert** »



*fig. 4* : suite du jeu après un clic sur « **Merge** »



*fig. 5* : le même jeu après 13 coups

(Examen de fin d'études secondaires : session de septembre 2009)