

Partie théorique : 20 points – 30 minutes

Question 1

11 (3+8) points

- (1) Exposer l'idée de l'algorithme du *tri par insertion*.
- (2) Ecrire une version récursive de cet algorithme.

Question 2

9 (=7+2) points

On considère la fonction suivante :

```
function p(s1,s2:string):integer;
begin
  result:=1;
  while (length(s1)<=length(s2)) and (s1<>copy(s2,1,length(s1))) do
    begin
      delete(s2,1,1);
      result:=result+1
    end;
  if length(s1)>length(s2) then result:=0
end;
```

- (1) Evaluer
 - a) `p('titi','titotititi')` ;
 - b) `p('tati','titato')` ;en suivant l'évolution des variables **s2** et **result** dans un tableau.
- (2) Expliquer ce que calcule la fonction **p** en général ?

Partie pratique : 40 points – 80 minutes

Le but de cette partie est d'implémenter dans Delphi une version partielle du jeu de dés « *Yahtzee* ». Le joueur y lance 5 dés dans le but de réaliser certaines *combinaisons* qui rapportent des points. Il y a 13 combinaisons possibles :

<i>Combinaison</i>	<i>Description</i>	<i>Points obtenus</i>
As	Dés avec 1 point	1 × le nombre de dés avec 1 point
Deux	Dés avec 2 points	2 × le nombre de dés avec 2 points
Trois	Dés avec 3 points	3 × le nombre de dés avec 3 points
Quatre	Dés avec 4 points	4 × le nombre de dés avec 4 points
Cinq	Dés avec 5 points	5 × le nombre de dés avec 5 points
Six	Dés avec 6 points	6 × le nombre de dés avec 6 points
Brelan	3 dés identiques	Somme des points sur les 5 dés
Carré	4 dés identiques	Somme des points sur les 5 dés
Full	3 dés identiques + 2 dés identiques	25 points
Petite suite	4 dés dont les chiffres se suivent	30 points
Grande suite	Les 5 dés se suivent	40 points
Yahtzee	5 dés identiques	50 points
Chance	Aucune condition	Somme des points sur les 5 dés

Pour réaliser *une* combinaison, le joueur a droit à *trois lancers au plus*, ce qui constitue *un tour* de la partie. A chaque clic sur le bouton « **Lancer** », il est libre de relancer tous les dés ou juste ceux de son choix, qu'il désigne par une croix à côté du dé.

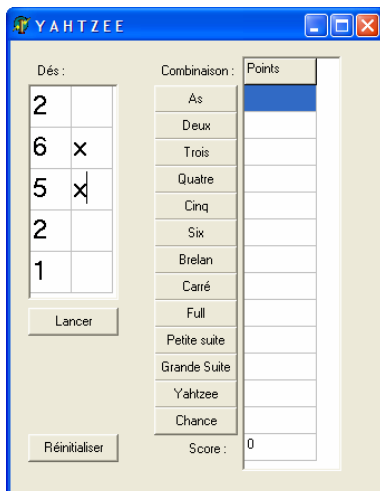


fig. 1

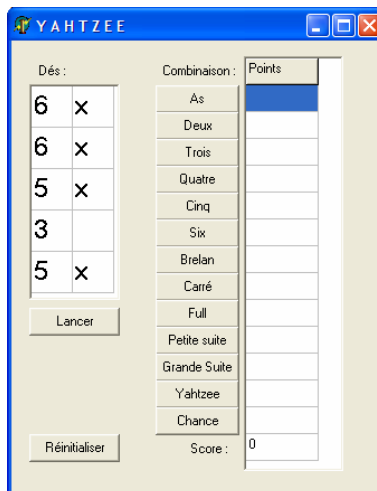


fig. 2

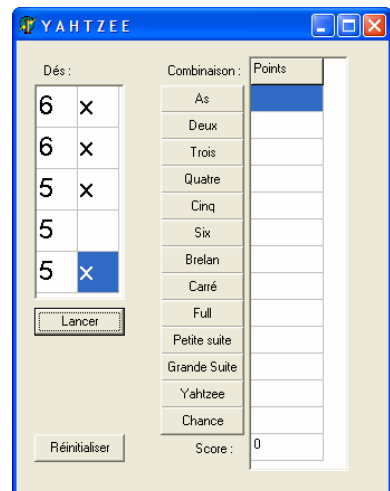
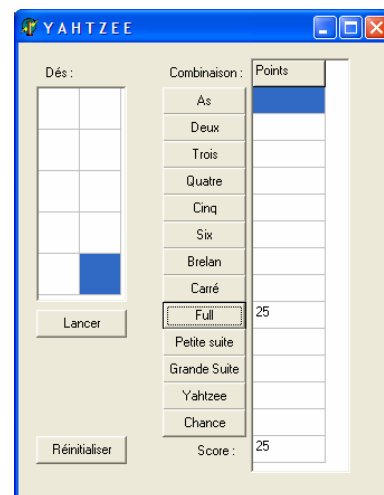


fig. 3

Exemple : Sur la figure 1, le joueur a cliqué une 1^{re} fois sur « **Lancer** ». Il décide de garder le 5 et le 6. Il clique ensuite une 2^e fois sur « **Lancer** » (2^e figure) et obtient un autre 6, un 3 et un autre 5. Il décide de garder les deux 5 et les deux 6 et lance une 3^e fois. Il obtient alors un troisième 5 (fig. 3) et a réalisé ainsi par exemple un « Brelan », un « Full » ou simplement un « Cinq » ou un « Six ».

A la fin d'un tour, le joueur doit *obligatoirement* cliquer sur *un* bouton avec une combinaison : si la combinaison a bien été réalisée, le score correspondant est affiché sur la feuille de score à côté du bouton, sinon le score 0 y est affiché. *Attention* : le joueur a le droit de cliquer *une seule fois* sur chaque combinaison !

Dans l'exemple ci-dessus, le joueur peut cliquer sur le bouton « **Full** », auquel cas il obtient 25 points. Si cette case de la feuille de score était déjà remplie, il pourrait toujours cliquer sur « **Brelan** », sur « **Cinq** », sur « **Six** », ou même sur « **Chance** ». Plus le jeu avance, moins il reste de possibilités pour faire des points. Supposons que notre joueur clique sur « **Full** ». Son score total augmente alors de 25 points, le tableau avec les résultats des dés est vidé et le joueur peut entamer le deuxième tour de la partie.



Programmation :

- (1) Créer l'interface graphique contenant notamment un *stringgrid* **sgD** (5 lignes et 2 colonnes) pour les dés et les croix, un *stringgrid* **sgScore** (15 lignes et 1 colonne) pour les scores, et les 13 boutons des combinaisons à côté des cellules correspondantes du tableau **sgScore**. (8 points)
- (2) Au démarrage de l'application, le générateur de nombres aléatoires sera initialisé, on écrira '**Points**' dans la ligne fixe du tableau **sgScore** et '**0**' (score total) dans la dernière ligne. Par ailleurs, pour contrôler le nombre de clics sur le bouton « **Lancer** », on utilisera une *variable globale* **nb_lancers** qui sera également initialisée à 0. (2 points)
- (3) Ecrire la méthode associée au clic sur le bouton « **Lancer** ». Si le joueur veut lancer le dé plus que 3 fois dans un tour, il sera averti par un message ! (6 points)
- (4) Ecrire les fonctions indépendantes :
 - a) **compter**, qui servira à compter le nombre de fois qu'un chiffre donné apparaît dans la première colonne du tableau **sgD** (4 points) ;
 - b) **nb_val_diff**, qui servira à compter le nombre de valeurs différents qui se trouvent dans la première colonne du tableau **sgD** (4 points) ;
- (5) Ecrire les méthodes associées aux boutons « **Six** » (4 points), « **Carré** » (5 points), et « **Full** » (5 points). Si le joueur essaie d'utiliser une même combinaison deux fois il devra être averti par un message !

G. Lorang