

## Partie écrite (22 points, durée : 30 min)

## Question 1

14 (7+5+2) points

- (1) Ecrire une procédure qui donne la primitive s'annulant en 0 d'un polynôme.
- (2) Ecrire une fonction qui évalue un polynôme en un réel donné par le procédé de Horner.
- (3) Expliquer comment les polynômes sont implémentés dans ces algorithmes.

## Question 2

8 points

On considère la fonction suivante :

```
function u(a, b: integer): integer;
begin
  if (a<=1) or (b<=1) then result:=1
  else if (a>b) then result:=a + u(a mod b, b)
  else if (b>a) then result:=b + u(a, b div a)
  else result:= a;
end;
```

Evaluer avec un calcul détaillé :

- (1)  $u(7,17)$
- (2)  $u(24,18)$

## Partie pratique (38 points, durée : 70 min)

Au jeu d'échecs, une dame menace toutes les cases des rangées horizontale, verticale et diagonales sur lesquelles elle se trouve. Sur la figure 1 ci-dessous par exemple, la dame (D) se trouve en D6 et menace toutes les cases marquées d'une croix (X). Le but de la programmation est de placer au hasard autant de dames sur l'échiquier que possible mais de façon à ce qu'elles ne se menacent pas mutuellement. Evidemment le nombre maximum de dames qu'on pourra placer est 8. Une partie est terminée lorsque toutes les cases de l'échiquier sont occupées ou menacées, comme sur la figure 2.

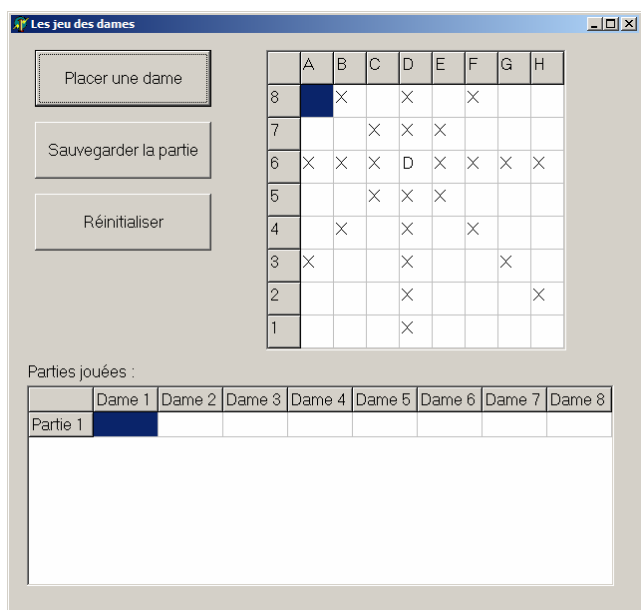


figure 1

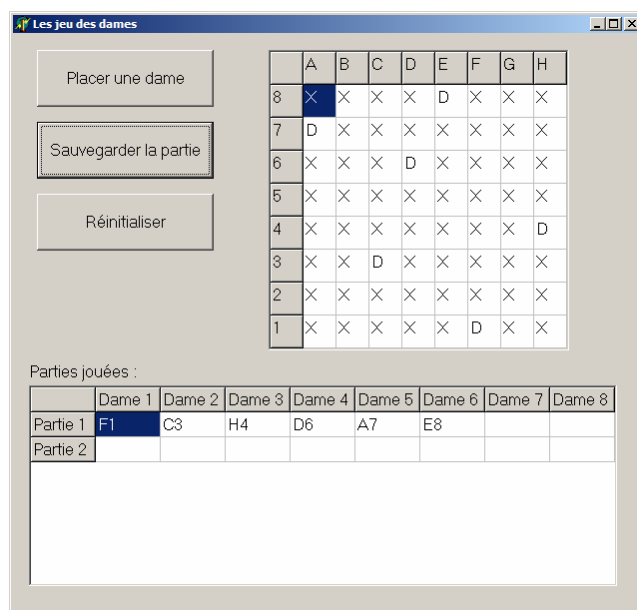


figure 2

- (1) Créer l'interface graphique de la figure 1, contenant notamment :
  - un *stringgrid* **sgD** (9x9) représentant l'échiquier,
  - un *stringgrid* **sgP** (2x9) servant à sauvegarder les parties jouées,
  - trois *buttons* **btnPlacerDame**, **btnSave** et **btnReinit** avec les *captions* respectives visibles sur la figure (1).

**Remarque** : Les deux *stringgrids* ont chacun une ligne et une colonne **fixes**, mais ne contiennent **aucun texte** avant l'exécution du programme. (4 pts)

- (2) Programmer la méthode « **Preparation** » associée à l'événement « **OnCreate** » du formulaire. Cette procédure a comme mission de remplir les lignes et colonnes fixes des deux *stringgrids* au moment de l'exécution du programme (comme sur la figure 1) et d'initialiser le générateur de nombres aléatoires. (8 pts)

- (3) Ecrire la *fonction* booléenne « *full* » qui retourne « *true* » si l'échiquier est entièrement rempli (de 'D' et de 'X', comme sur la figure 2) et « *false* » sinon.  
(4 pts)
- (4) Un clic sur le bouton « *Placer une dame* » aura comme effet de placer *au hasard* une dame 'D' sur une *case vide* de l'échiquier et de marquer par une croix 'X' toutes les cases menacées par cette dame. Lorsque toutes les cases de l'échiquier sont occupées par 'D' ou 'X', l'utilisateur sera averti par un message.  
(10 pts)
- (5) Un clic sur le bouton « *Sauvegarder la partie* » aura comme effet de lire *ligne par ligne* les cases de l'échiquier *en commençant en bas* (rangée numérotée 1) et de sauvegarder les positions des dames dans le *stringgrid sgP* (voir figure 2). Le *stringgrid sgP* sera également adapté afin de pouvoir contenir le résultat de la partie suivante (voir figure 2). Lorsque l'utilisateur voudra sauvegarder une partie non terminée, il sera averti grâce à un message adéquat. (10 pts)
- (6) Un clic sur le bouton « *Réinitialiser* » aura comme effet de vider l'échiquier. (L'utilisateur cliquera sur ce bouton avant de commencer une nouvelle partie.)  
(2 pts)

*N.B. : La correction tiendra compte du style de programmation et de la lisibilité du code.*

G. Lorang