

Epreuve écrite

Examen de fin d'études secondaires 2013

Section: B

Branche: Informatique

Numéro d'ordre du candidat

Partie théorique (50 minutes - 30 points)

Question 1 (10 points)

- (1) Exposer l'idée de l'algorithme du *tri par sélection* dans l'ordre croissant d'une liste. **(2 points)**
- (2) Ecrire la version itérative de cet algorithme de tri (avec la procédure auxiliaire) **(6 points)**
- (3) On trie la liste $lbEx := ('C', 'D', 'B', 'A')$ à l'aide de l'algorithme précédent. Quels changements successifs cette liste subit-elle au cours de l'exécution ? **(2 points)**

Question 2 (12 points)

- (1) Ecrire la fonction **produit** qui retourne le produit de deux polynômes. Préciser comment les polynômes sont implémentés dans cette fonction. **(6 points)**
- (2) En déduire une **fonction récursive puissance**, qui prend en entrée un polynôme **p** et un entier naturel **n** et qui retourne la n^e puissance du polynôme **p**. On demande d'adapter l'algorithme « *rapide* » du cours vu dans le cas du calcul d'une puissance d'un réel. (N.B. : on définira $p^0(x) = 1$, comme dans le cas des réels.) **(6 points)**

Question 3 (8 points)

On donne la fonction booléenne récursive suivante :

```
function mystere(a,b :string):boolean ;
begin
  if (a='') then result := true
    else if pos(a[1],b)=0 then result := false
      else begin
        delete(a,1,1) ;
        result := mystere(a,b)
      end
    end ;
end ;
```

- (1) Calculer en précisant toutes les étapes :
 - a) `mystere('musique', 'mathematiques')`
 - b) `mystere('musicien', 'instrument')` **(4 points)**
- (2) Donner trois exemples de strings **x** avec des longueurs différentes tels que la condition `mystere(x, 'abc')` and `mystere('abc', x)` soit égale à `true`. **(2 points)**
- (3) Expliquer ce que calcule la fonction `mystere` en général. **(2 points)**

Epreuve écrite

Examen de fin d'études secondaires 2013

Section: **B**

Branche: **Informatique**

Numéro d'ordre du candidat

Partie pratique : My Music (70 minutes - 30 points)

Il s'agit de créer une application Delphi qui permet à l'utilisateur d'ajouter ou d'effacer des morceaux de musique à une base de données et de créer ensuite un CD avec des morceaux sélectionnés aléatoirement dans cette collection.

- (1) Créer le formulaire ci-dessous en respectant les dénominations de la figure ; le tableau **sgM** et les boîtes d'édition sont initialement vides. Le nombre de colonnes de **sgM** est fixé à 5 (dont une colonne fixe). Le nombre de lignes est initialement égal à 3 (dont une ligne fixe) et variera au cours de l'exécution du programme. Le libellé **lblNumber** (valeur par défaut : 0) devra afficher le nombre de morceaux exact dans la collection tout au long de l'exécution. (4 points)

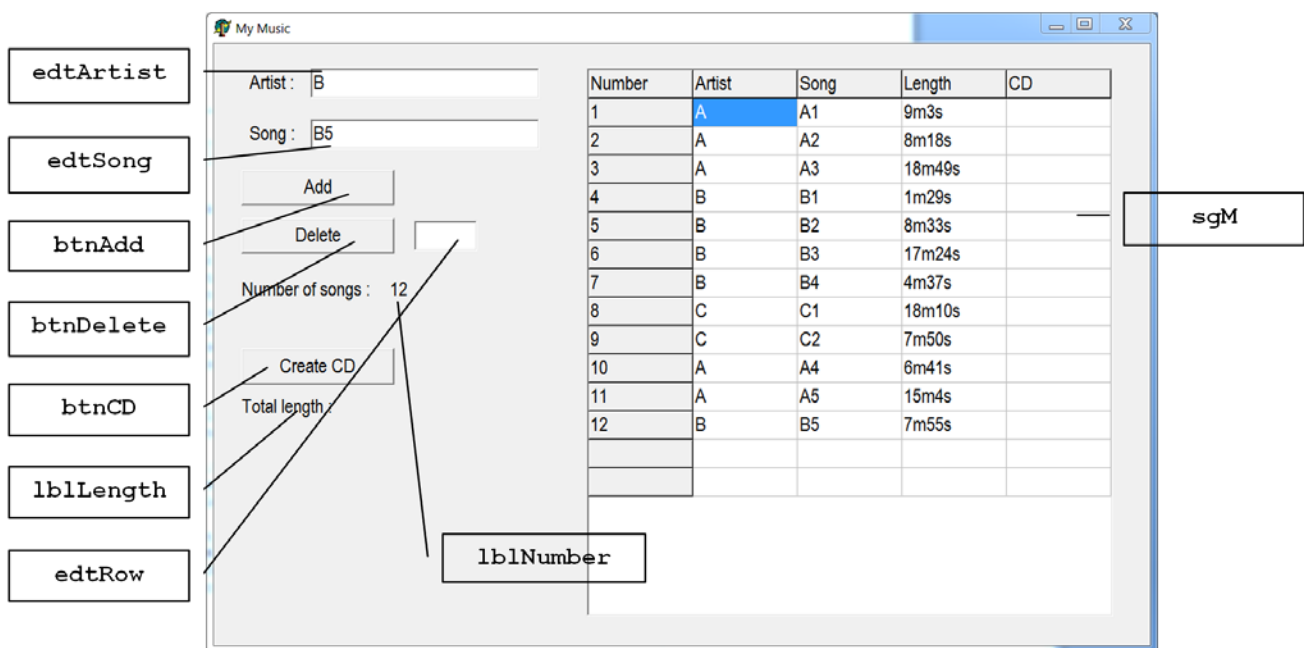


fig. 1 (sur laquelle on vient d'ajouter le 12^e morceau à la collection)

- (2) Ecrire la méthode **FormCreate** qui doit initialiser le générateur de nombres aléatoires et la ligne fixe du tableau **sgM** comme sur la fig. 1 au démarrage du programme. (2 points)
- (3) Un clic sur le bouton **Add** permet d'ajouter un nouveau morceau de musique à la collection : le nombre de lignes de **sgM** et le libellé **lblNumber** augmenteront de 1, les noms de l'artiste et du morceau, que l'utilisateur aura écrits respectivement dans les boîtes d'édition **edtArtist** et **edtSong**, seront transférés dans la 1^{re} ligne encore vide du tableau **sgM** et le numéro correct du morceau sera ajouté dans la colonne *Number*. La durée du morceau sera choisie *au hasard* par le programme et ajoutée dans la colonne *Length*. Elle sera affichée dans le format '**xmys**' où **x** est le nombre de minutes compris entre 0 et 19 (bornes incluses) et **y** est le nombre de secondes comprises entre 0 et 59 (voir la figure 1, sur laquelle le 12^e morceau vient d'être ajouté à la collection). (5 points)

Epreuve écrite

Examen de fin d'études secondaires 2013

Section: B

Branche: Informatique

Numéro d'ordre du candidat

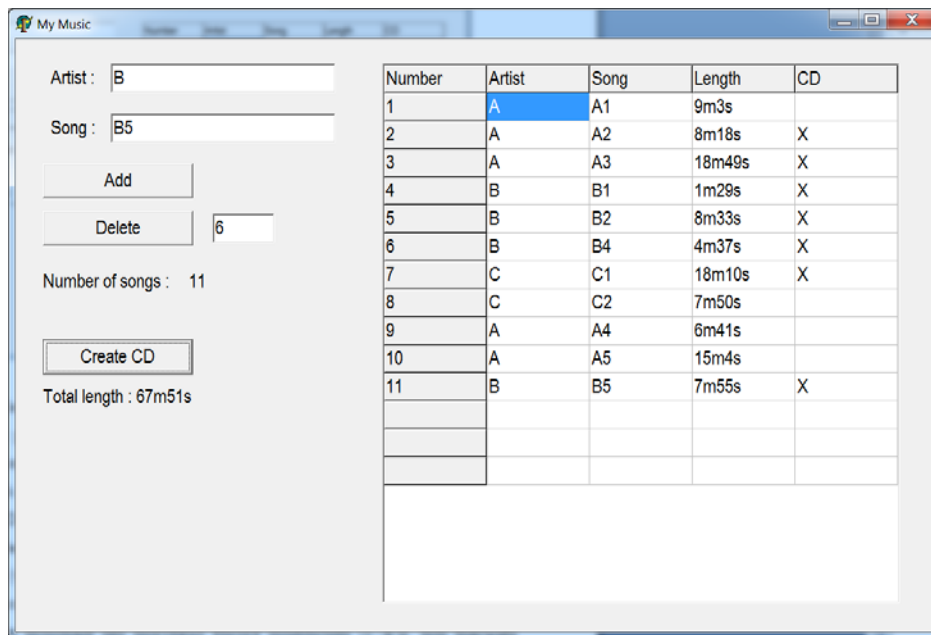


fig. 2 (sur laquelle on a effacé le morceau B3 de la fig. 1, puis lancé la création du CD)

- (4) Un clic sur le bouton **Delete** permet à l'utilisateur d'effacer le morceau dont il a écrit le numéro dans la boîte d'édition **edtRow**. Si l'utilisateur essaie d'effacer un morceau avec un numéro non valide, il en sera informé par un message. Le nombre de morceaux de la collection **lblNumber** diminuera de 1 et les morceaux seront renumérotés de 1 à **lblNumber**. Par exemple, sur *la fig. 2* on peut observer l'effet produit par l'effacement du morceau numéro 6 de la *fig. 1*. **(5 points)**
- (5) Lorsque l'utilisateur clique sur le bouton **Create CD**, le programme sélectionnera aléatoirement des morceaux de musique qui pourront être gravés par la suite sur un CD. Le programme mettra une croix dans la dernière colonne du tableau **sgM** pour chaque morceau qu'il aura sélectionné. Les contraintes à respecter sont les suivantes :
- le CD comporte au plus 74 minutes de musique, c.-à-d. 4440 secondes. La durée totale du CD sera ajoutée au libellé **lblLength** dans le format '**xmys**' ;
 - le programme continuera à sélectionner au hasard des morceaux jusqu'à ce que, soit tous les morceaux aient été choisis (sans que le CD soit plein), soit la durée minimale des morceaux encore disponibles dépasse le temps restant sur le CD (voir *fig. 2*).

Le programme comportera obligatoirement a) une fonction **toSeconds** qui transforme un string donné au format '**xmys**' en le nombre de secondes correspondant et b) une fonction **min** qui retourne le morceau le plus court non encore sélectionné de la collection. **(14 points)**