27/07/2016

MEKSAVANH Teddy, BERANGER Aurélien, LENFANT Valentin

Groupe 10

Rapport de projet : groupe 10

**SOMMAIRE**

* Introduction
* Notice fonctionnelle
* Algorithme principal
* Conclusion

Introduction : ce projet regroupe 2 grandes parties : une partie « répertoire » au sein de laquelle il est possible d’effectuer les actions classiques que l’on retrouve dans ce genre d’applications, à savoir :

* Ajout d’un nouveau contact
* Modification d’un contact existant
* Suppression d’un contact existant
* Affichage des contacts

Mais également une partie « cryptographie » qui permet de rendre les données illisibles à moins de décrypter le fichier contenant les contacts. Il est ainsi possible de :

* Crypter les contacts
* Décrypter les contacts

Notice fonctionnelle : une fois l’application exécutée, vous arriverez dans un menu vous proposant 7 choix différents. Chaque choix est explicité et peut-être exécuté individuellement en choisissant un nombre entre 1 et 7. Toute autre entrée différente d’un nombre compris entre 1 et 7 [1 ; 7] retournera un message d’erreur. Celui-ci vous invitera à refaire votre choix.

**L’action numéro 1** vous permet d’ajouter un nouveau contact à votre carnet. Pour ce faire, l’application va vous guider pas à pas dans l’ajout d’un nouveau contact. Dans l’ordre, l’application vous demandera les informations suivantes que vous devrez saisir :

* Nom
* Prénom
* Numéro de l’habitation
* Type de la voie
* Nom de la voie
* Code postal
* Ville
* Email
* Téléphone

Une fois l’ensemble de ces informations saisies, l’application vous demandera si vous souhaitez saisir un autre contact. Vous pouvez soit répondre « non » en rentrant 0 ou répondre « oui » en rentrant 1. Une saisie invalide entrainera l’affichage d’un message d’erreur vous demandant de refaire votre choix.  
En cas de réponse positive, vous devrez de nouveau saisir toutes les informations indiquées précédemment et en cas de réponse négative, l’application va vous demander si vous souhaitez retourner dans le menu. En cas de réponse positive, vous pourrez de nouveau exécuter une action parmi les 7 présentes. En cas de réponse négative, l’application se terminera.

**L’action numéro 2** vous permet de modifier un contact existant de votre carnet. Pour ce faire, l’application va vous lister dans un premier temps l’ensemble des contacts présents dans votre répertoire. Vous pourrez ensuite choisir quel contact vous souhaitez modifier en entrant son numéro associé, chaque contact étant identifié de manière unique par un numéro.  
Une fois votre choix fait, l’application vous demandera quel champ vous souhaitez modifier. Là encore, chaque information est représenté par un numéro allant de 1 à 9. Les 9 choix sont les suivants :

* Nom
* Prénom
* Numéro de l’habitation
* Type de la voie
* Nom de la voie
* Code postal
* Ville
* Email
* Téléphone

Quel que soit le champ que vous souhaitez modifier, l’application vous demandera de saisir la nouvelle valeur à appliquer. Une fois la nouvelle valeur saisie et validée, l’application vous demandera si vous souhaitez modifier un autre champ du même contact. En cas de réponse positive, les 9 choix décrits précédemment seront de nouveau proposé. En cas de réponse négative, l’application vous demandera si vous souhaitez modifier un autre contact. En cas de réponse positive, la liste de l’ensemble des contacts modifiables s’affichera de nouveau et en cas de réponse négative, l’application vous demandera si vous souhaitez retourner dans le menu.

**L’action numéro 3** vous permet de supprimer un contact existant de votre carnet. Pour ce faire, l’application va vous lister dans un premier temps l’ensemble des contacts présents dans votre répertoire. Vous pourrez ensuite choisir quel contact vous souhaitez supprimer en entrant son numéro associé. Une fois le contact supprimé, l’application va vous demander si vous souhaitez supprimer un autre contact. En cas de réponse positive, la liste des contacts restant sera affichée et vous pourrez de nouveau faire votre choix. En cas de réponse négative, l’application va vous demandez si vous souhaitez retourner dans le menu.

**L’action numéro 4** vous permet d’afficher l’ensemble des contacts présent dans votre répertoire. Suite à cela, l’application vous demandera si vous souhaitez retourner dans le menu.

**L’action numéro 5** vous permet de crypter l’ensemble des contacts de votre répertoire afin de rendre les informations illisibles par un humain.

**L’action numéro 6** vous permet de décrypter le cryptage préalablement effectué sur votre répertoire afin de rendre les informations lisibles par un humain.

**L’action numéro 7** vous permet tout simplement de quitter le programme.

Algorithme principal : l’algorithme principal présenté sera celui servant à crypter les informations contenues dans le répertoire. L’algorithme sera présenté en pseudo code afin de présenter de manière claire le fonctionnement et l’ensemble des mécaniques utilisées par la partie cryptographie du programme.

Procédure detruireCleDeDecryptage ()  
{  
 //nom du fichier de decryptage est le nom du fichier servant à décrypter les contact  
 Supprimer(« nom du fichier de decryptage »)   
}

---------------------------------------------------------------------------------------------------------------------------------

Procédure cryptageDuFichier (caractereEnCours, premiereBorne, deuxiemeBorne)  
{  
 //fichier et fichier2 sont de type FILE\* à savoir un pointeur sur un fichier  
 POINTEURSURFICHIER fichier = null  
 POINTEURSURFICHIER fichier2 = null  
  
 //On ouvre le fichier contenant les contacts en lecture seule  
 fichier = fopen(« Contact.txt », « r »)  
 //On ouvre le fichier qui contiendra les contacts en écriture  
 fichier2 = fopen(« ContactCrypte.txt », « w »)  
 //On vérifie que le fichier est bien ouvert  
 SI (fichier != null)  
 {  
 FAIRE  
 {  
 //On met dans caractereEnCours le caractère pointé dans fichier  
 caractereEnCours = fgetc(fichier)  
 //On écrit dans ContactCrypte.txt le nouveau caractère crypté  
 fputc (applicationCryptageCaractere(caractereEnCours, premiereBorne, deuxiemeBorne) fichier2)  
 }   
 //EOF pour End Of File représente la fin du fichier  
 TANT QUE (caractereEnCours != EOF)  
 //On ferme les 2 fichiers  
 fclose(fichier)  
 fclose(fichier2)  
 }  
}

---------------------------------------------------------------------------------------------------------------------------------

Fonction applicationCryptageCaractere (caractereEnCours, premiereBorne, deuxiemeBorne)  
{  
 ENTIER choixDuCryptage = fonctionChoixDuCryptage(premiereBorne, deuxiemeBorne)  
 SUIVANT (choixDuCryptage)  
 {  
 cas 1 :   
 ecritureFichierDecryptage(choixDuCryptage)  
 caractereEnCours = unAlgoDeCryptage(caractereEnCours)  
 fin cas 1  
 cas 2  
 ecritureFichierDecryptage(choixDuCryptage)  
 caractereEnCours = unAlgoDeCryptage(caractereEnCours)  
 fin cas 2  
 }  
 RETOURNER caractereEnCours  
}

---------------------------------------------------------------------------------------------------------------------------------

Fonction fonctionChoixDuCryptage (premiereBorne, deuxiemeBorne)  
{  
 RETOURNER aleatoire()MOD(deuxiemeBorne – premiereBorne) + premiereBorne  
}

---------------------------------------------------------------------------------------------------------------------------------

Procédure ecritureFichierDecryptage (choixDuCryptage)  
{  
 POINTEURSURFICHIER fichier2 = null  
 fichier2 = fopen(« decryptage.txt », « a »)  
 SI (fichier2 != null)  
 {  
 // Ecriture du nombre contenu dans choixDuCryptage qui servira à la clé de décryptage  
 fprintf(fichier2, « %d », choixDuCryptage)  
 fclose(fichier2)  
 }  
}

Conclusion : plusieurs pistes d’améliorations potentielles pourraient être envisagées. Tout d’abord, nous pourrions doter le programme d’une interface graphique. Ensuite, nous pourrions optimiser la saisie d’un nouveau contact en ne décomposant pas l’adresse d’une personne. Des vérifications sur la saisie des différents champs pourraient également être mises en place.