Projet
répertoire téléphonique

Un répertoire téléphonique doit contenir au minimum des numéros d'appel ainsi qu'un nom les identifiant.
Il est possible d'ajouter des informations supplémentaires, par exemple :

- prénom,
- tel fixe et portable (soit deux numéros),
- adresse e-mail,
- ...

En C nous utiliserons les structures pour facilité l'utilisation des données. Avec les données précédentes il est fort possible de n'utiliser que des tableaux de char pour chacune d'entre elles cependant pour le numéro de téléphone fixe et portable, le type uint64_t peut être utilisé; c'est un type pouvant prendre un entier non signé (unsigned : toujours positif) d'une longueur de 64 bits. Le uint64_t se trouve dans l'header stdint.h c'est un type standardisé donc portable. Pour se faire il faudra convertir un numéro de téléphone du type char* (une chaîne) au type uint64_t et donc créer une fonction permettant cette conversion.

```
struct contact{
       char *nom, *prenom, *email;
        uint64_t tel_fix, tel_mob;
};
```

Nous avons donc la structure contact. Les chaînes de caractères nom, prénom, et email seront allouées dynamiquement pour ne pas restreindre leur taille. La taille des chaînes sera donc fixée et comptée à l'entrée de l'utilisateur et utilisée pour l'allocation dynamique d'un tableau de char dont l'adresse sera simplement passée dans la structure.

Il faudra créer des fonctions pour simplifié tout ceci mais, exemple :

```
char buff[256];
```

L'utilisateur entre un nom et on le met dans le tableau buff. Ensuite on calcule la taille du nom contenu dans buff :

```
int n;
n = strlen(buff);
```

et on utilise la taille du nom pour alloué un bloc mémoire et on met l'adresse de ce bloc mémoire dans le pointeur *nom de la structure.

```
struct contact ct;
ct.nom = malloc(sizeof(char)*n+1);
```

puis on copie la chaîne dans le bloc alloué :

```
strcpy(ct.nom,buff);
```

Ceci devra être simplifié par une fonction.

Quelque fonctions à créer pour le projet :

La conversion d'un nombre se trouvant dans une chaîne de caractères en uint64_t (en entier). Elle prendra donc une chaîne de caractères en paramètre et retournera un uint64_t.
Le prototype de la fonction sera le suivant :

```
uint64_t stru64(char *str);
```

Son utilisation sera de la forme :

```
uint64_t a;
a = stru64("0123456789");
```

La récupération d'une phrase tapée par l'utilisateur en console. La fonction devra utiliser la fonction getchar() de l'header stdio.h , cette fonction sert à la récupération d'un caractère tapé en console et utilisé dans une boucle elle permet donc de récupérer plusieurs caractères à la suite utile pour les noms et prénoms composés (contenant un espace).
On précisera la taille max pour ne pas déborder le tableau.
Le prototype de la fonction sera le suivant :

```
int getstr(char *buff,int max);
```

Son utilisation sera de la forme :

```
char buff[256];
int n;
n = getstr(buff,256);
```

L'allocation dynamique d'une chaîne de caractère et le retour de l'adresse.
La fonction prendra en paramètre le buffer contenant la chaîne à copier et retournera la nouvelle adresse de la chaîne.

```
char *newstr(char *buff);
```

Son utilisation sera de la forme :

```
char buff[256]="Dupont";
char *new;
new = newstr(buff);
printf("%s",new);
```

Devra afficher Dupont.