

<p>Bases de Données TP 9a : PRO*C Sélection Mono - Tuple</p>
--

- (1) Copier dans votre répertoire courant le fichier 'debut.pc' contenant un programme pro*c permettant une connexion à votre base de données.
- (2) Etudier ce programme source.
- (3) Compiler ce programme avec la commande :

```
make -f proc.mk EXE=debut OBJS=debut.o
```
- (4) Etudier le programme debut.c généré.
- (5) Exécuter le programme binaire 'debut'.
- (6) Reprendre le programme 'debut.pc' et créer le programme 'tp12.pc' en ajoutant le code nécessaire permettant d'afficher les renseignements concernant un projet dont le numéro est entré au clavier.

Programmer la fonction 'selection()' avec la méthode générale suivante :

```

1- Entrer un numéro de projet ou 'fin'
2- Si 'fin aller en 5
3- Afficher :
Nom du projet      Nom de l'équipe      Nom chercheur responsable
-----
ou
Nom du projet      Nom de l'équipe      Pas de chercheur responsable
-----
ou
                    'Projet inconnu'
4- Entrer un numéro de projet ou 'fin', aller en 2
5- Fin de la sélection

```

- (7) Compiler le programme tp12.pc et l'exécuter pour tester. Exécuter le programme en conservant la trace écran dans un fichier 'script' unix (équivalent du SPOOL de sqlplus) :

(8)

```

script tp12.log
.... exécution du tp12 avec les 3 cas possibles .....
<ctrl><D>

```

- (9) Résultats à rendre :

Un seul listing contenant :

- contenu du programme source 'tp12.pc'
- trace de l'exécution (contenu du fichier script)

```
/* debut.pc PRO*C : CONNECTION ET DECONNEXION , SQUELETTE PROGRAMME
*/
```

```
#include <stdio.h>
#include <ctype.h>
#include <string.h>
```

```
EXEC SQL BEGIN DECLARE SECTION;
    VARCHAR user[20];
    VARCHAR pwd[20];
EXEC SQL END DECLARE SECTION;
```

```
EXEC SQL INCLUDE sqlca.h;
```

```
connection()
```

```
{
    /* entree du mot de passe et du nom */

    user.len=asks("Entrer votre user : ",user.arr);
    pwd.len=asks("Entrer votre pwd : ",pwd.arr);

    /* connection a la base */
    EXEC SQL WHENEVER SQLERROR GOTO erreur;
    EXEC SQL CONNECT :user IDENTIFIED BY :pwd;
    printf("\n Connexion reussie ..... \n");
    return(1);
erreur :
    printf("\nErreur a la connection .... ");
    errpt();
    return(0);
    EXEC SQL WHENEVER SQLERROR continue;
}
```

```
selection()
```

```
{
    /* fonction de selection */
    printf("\n Debut de selection ..... \n");
    printf("\n Fin de selection ..... \n");
}
```

```
main()
```

```
{
if (connection())
{
    selection();
    sortie();
}
else
    arret();
}
sortie()
```

```

{
EXEC SQL COMMIT RELEASE;
}
arret()
{
EXEC SQL ROLLBACK RELEASE;
}
errprt()
{
printf("\n%.70s (%d)\n",sqlca.sqlerrm.sqlerrmc, -sqlca.sqlcode);
}

```

```

int askn(text,variable)
char text[];
int *variable;
{
char s[20];
printf(text);
if ( gets(s) == (char *)0 )
return(EOF);
*variable = atoi(s);
return(1);
}

```

```

int asks(text,variable)
char text[],variable[];
{
printf(text);
return( gets(variable) == (char *)0 ? EOF : strlen(variable) );
}

```