

M3101 : Principes des systèmes d'exploitation

TD : Corrigé des exercices de C

TD1 les fonctions

Exercice 1

Question 1

```
#define LG_MAX 100

typedef enum {FAUX=0, VRAI=1} Booleen ;

typedef int TabEntiers[LG_MAX] ;

void rechercherOccurrence(const TabEntiers tab, const int n, const int ele,
                          Booleen * trouve, int * rang) ;
```

Question 2

```
void rechercherOccurrence(const TabEntiers tab, const int n, const int ele,
                          Booleen * trouve, int * rang)
{
    Booleen fini ;
    int i ;

    i = 0 ;
    fini = FAUX ;
    while (! fini)
    {
        if (i < n)
        {
            if (ele == tab[i])
            {
                * trouve = VRAI ;
                * rang = i ;
                fini = VRAI ;
            }
            else
            {
                i++ ;
            }
        }
        else
        {
            * trouve = FAUX ;
            fini = VRAI ;
        }
    }
}
```

Question 3

```
int main (void)
{
    TabEntiers listeNotes ;
    int nbreNotes ;
    int noteZero = 0 ;
    int rangNote ;
    Booleen trouve ;

    .....

    /* rechercher la note 0 */

    rechercherOccurrence(listeNotes, nbreNotes, noteZero, &trouve, &rangNote);

    .....
}
```

Exercice2

Question 1

```
typedef enum {FAUX=0, VRAI=1} Booleen ;

typedef int * PtrEntier ;

void rechercherSousTableau (const PtrEntier adeb, const PtrEntier afin,
                           const int ele, Booleen * trouve,PtrEntier * arang);
```

Question 2

```
void rechercherSousTableau (const PtrEntier adeb, const PtrEntier afin,
                           const int ele, Booleen * trouve,PtrEntier * arang)
{
    Booleen fini ;
    PtrEntier acour ;

    acour = adeb ;
    fini = FAUX ;
    while (! fini)
    {
        if (acour <= afin)
        {
            if (ele == * acour)
            {
                * trouve = VRAI ;
                * arang = acour ;
                fini = VRAI ;
            }
            else
            {
                acour++ ;
            }
        }
        else
        {
            * trouve = FAUX ;
            fini = VRAI ;
        }
    }
}
```

Question 3

```
int main (void)
{
    PtrEntier premiereNote ;
    PtrEntier derniereNote ;
    int noteZero = 0 ;
    Booleen trouve ;
    PtrEntier refNote ;

    .....

    /* rechercher la note 0 */

    rechercherSousTableau(premiereNote, derniereNote, noteZero, &trouve, &refNote);

    .....
}
```

TD2 : allocation dynamique

Question 1

```
/* declarations */

#include <stdlib.h>
#include <stdio.h>

#define LG_MAX    30

typedef int * PtrEntier;
PtrEntier tabVal;

/* allocation memoire dynamique du tableau d'entiers */

tabVal=(PtrEntier)malloc(LG_MAX * sizeof(int));
if(tabVal == NULL)
{
    /* afficher un message d'erreur et arreter le processus */
    perror("echec malloc");exit(1);
    /* ou lever une exception (sera vu au TD4) */
}

/* exemple d'utilisation du tableau */

tabVal[i]= -25;
```

Question 2

```
/* declarations */

#include <stdlib.h>
#include <stdio.h>

#define LG_MAX    256

typedef char * PtrCar;
PtrCar tampon;
```

```
/* allocation memoire dynamique du tableau de caractères */

tampon=(PtrCar)malloc(LG_MAX+1);
if(tampon == NULL)
{
    /* afficher un message d'erreur et arreter le processus */
    perror("echec malloc");exit(1);
    /* ou lever une exception (sera vu au TD4) */
}

/* exemple d'utilisation du tableau */
/* saisie d'une chaine de caracteres au clavier */

fgets(tampon,LG_MAX+1,stdin);
```

Question 3

```
/* declarations */

#include <stdlib.h>
#include <stdio.h>
#include <string.h>

typedef char TabCar60[60+1];

typedef struct
{
    int codeProd;
    TabCar60 designation;
    float puHT;
} EnregProd;

typedef EnregProd * PtrEnregProd;

PtrEnregProd unProduit;

/* allocation memoire dynamique de l'enregistrement unProduit */

unProduit=(PtrEnregProd)malloc(sizeof(EnregProd));
if(unProduit == NULL)
{
    /* afficher un message d'erreur et arreter le processus */
    perror("echec malloc");exit(1);
    /* ou lever une exception (sera vu au TD4) */
}

/* exemples d'utilisation de l'enregistrement unProduit */

unProduit->codeProd=12345;

(* unProduit).puHT=20.45;

strcpy(unProduit->designation, "chocolat poudre YABON 250g");
```

TD3 : les E/S sur fichiers

```
/* complements de C */
/* TD3-exo1 E/S sur fichier texte */

#include <unistd.h>
#include <stdlib.h>
#include <stdio.h>
#include <string.h>

typedef char TabCar25[25];

void saisirChaine(TabCar25 t);

/* programme principal */
void main()
{
    TabCar25 tab1,tab2,tab3;
    FILE * fic;

    /* ouverture du fichier*/
    fic=fopen("annuaire.txt", "w");
    if (fic == NULL)
        {perror("echec fopen"); exit(1);}

    /* saisie des donnees */
    printf("\nEntrer un nom ou stop: ");
    saisirChaine(tab1);

    while (strcmp(tab1,"stop") != 0)
    {
        printf("entrer un prenom: ");
        saisirChaine(tab2);
        printf("entrer un n° de telephone: ");
        saisirChaine(tab3);
        fprintf(fic,"%-25s%-25s%-15s\n",tab1, tab2, tab3);
        printf("\nEntrer un nom ou stop: ");
        saisirChaine(tab1);
    }
    fclose(fic);
}

void saisirChaine(TabCar25 t)
{
    int dernier;

    fgets(t,25,stdin);
    dernier=strlen(t)-1;
    if (t[dernier] == '\n')
        { t[dernier] = '\0';}
}

/* fin du source */
```

```
* complements de C */
/* TD3-exo2 E/S sur fichier binaire */
```

```
#include <unistd.h>
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
```

```
typedef char TabCar25[25];
typedef struct
{
    TabCar25 nom;
    TabCar25 prenom;
    TabCar25 telephone;
}EnregAnnu;
```

```
void saisirChaine(TabCar25 t);
```

```
/* programme principal */
void main()
{
    TabCar25 tab1,tab2,tab3;
    EnregAnnu infoAnnu;
    FILE * fic;
```

```
/* ouverture du fichier*/
fic=fopen("annuaire.bin", "w");
if (fic == NULL)
    {perror("echec fopen"); exit(1);}
```

```
/* saisie des donnees */
printf("\nEntrer un nom ou stop: ");
saisirChaine(tab1);
```

```
while (strcmp(tab1,"stop") != 0)
{
    printf("entrer un prenom: ");
    saisirChaine(tab2);
    printf("entrer un n° de telephone: ");
    saisirChaine(tab3);
    strcpy(infoAnnu.nom,tab1);
    strcpy(infoAnnu.prenom,tab2);
    strcpy(infoAnnu.telephone,tab3);
    fwrite(& infoAnnu, sizeof(EnregAnnu), 1, fic);
    printf("\nEntrer un nom ou stop: ");
    saisirChaine(tab1);
}
fclose(fic);
}
```

```
void saisirChaine(TabCar25 t)
{
    int dernier;
```

```
fgets(t,25,stdin);
dernier=strlen(t)-1;
if (t[dernier] == '\n')
    { t[dernier] = '\0';}
}
```

```
/* fin du source */
```

TD4 : les points de reprise

```
/* complements de C */
/* TD4-ex01 points de reprise */

#include <unistd.h>
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <setjmp.h>

typedef char TabCar40[40];

void saisirChaine(TabCar40 t);
FILE* ouvrirFichier(TabCar40 df, jmp_buf pr);

/* programme principal */
void main()
{
    int ret,cpt;
    int nbEssai=3;
    jmp_buf ptRep;
    TabCar40 desFic;
    FILE * fic;

    /* pose point de reprise*/
    ret=setjmp(ptRep);
    if (ret == 0)
        {cpt=0;}
    else
    {
        perror("echec fopen");
        sleep(5);
        if (cpt >= nbEssai)
        {
            printf("*** Echec de l'ouverture du fichier: %s\n", desFic); exit(1);
        }
    }

    cpt ++;
    /* saisie de la designation du fichier */
    printf("\nEnter la designation du fichier: ");
    saisirChaine(desFic);

    /* ouverture du fichier */
    fic=ouvrirFichier(desFic, ptRep);

    printf("*** Ouverture correcte du fichier:  %s\n", desFic);
    fclose(fic);
}

void saisirChaine(TabCar40 t)
{
    int dernier;

    fgets(t,40,stdin);
    dernier=strlen(t)-1;
    if (t[dernier] == '\n')
        { t[dernier] = '\0';}
}
```

```
FILE* ouvrirFichier(TabCar40 df, jmp_buf pr)
{
    FILE* fic;
    fic=fopen(df, "r");
    if (fic == NULL)
        {
            longjmp(pr,1);
        }
    return fic;
}
/* fin du source */
```