

Exercice I

1. Tracer l'état de mémoire du programme ci-dessous :

```
#include <stdio.h>

int f(int*,int);

int main(void){
    int t[]={2,3,4};
    int z=f(t,3);
    return 0;
}

int f(int *t,int s){
    int tmp,z;
    if(s<=0) return 0;
    if(s==1) return *t;
    tmp=*t;
    *t=*(t+1);
    *(t+1)=tmp;
    return z=*t*f(t+1,s-1);
}
```

Exercice II

2. Définir la fonction printR qui imprime sur l'écran une chaîne de caractères donnée dans le sens inverse. On ne peut pas utiliser les fonctions de la bibliothèque "string.h". La version récursive sera privilégiée, mais pas obligatoire. Exemple: printR("animal"); affiche sur l'écran : lamina
3. Définir la fonction printReverse qui imprime les chaînes dans un tableau dans le sens inverse. Cette fonction utilise la fonction printR. On donne ci-dessous, la fonction printReverseTest() et le résultat de l'exécution :

```
void printReverseTest(){
    char *t[]={"animal","desserts",NULL,"gateman","", "\0", "depots"};
    printReverse(t,7);
}

int main(void) { printReverseTest(); return 0; }
```

Résultat affiché sur l'écran (_ indique espace) :
stoped__nametag__stressed_lamina_

Exercice III

Considérer le type cel : typedef struct cel { int d; struct cel *suiv;} cel;

4. Définir la fonction extraire qui extrait une sous-liste à partir d'une liste donnée d'entiers. Les cellules devant être extraites sont celles contenant un entier donné. Aucune allocation ou libération n'est permise.

Exemple:

```
cel *L1, *L2=NULL;
//ajouter (Push) des cellules pour construire la liste {2,1,2,4,2,2}
extraire(&L1,&L2,2);
```

