```
// TD/TP 7

// exercice 1. fonction récursive qui calcule la fonction nième puissance de x ,
// où x est un réel et n un entier positif ou nul.

float puiss(float x, int n){ // retourne 0 si n négatif
  if (n<0) return 0;
  if (n==0) return 1;
  else return x * puiss(x,n-1) ;
}

// exercice 2. une fonction récursive qui calcule 1+x+...puiss(x,n).

float somme_puiss(float x, int n){ // retourne 0 si n négatif ou nul
  if (n<=0) return 0 ;
  return x * somme_puiss(x,n-1)+ 1 ;
}

// exercice 3.

int suite(int n){ // retourne 0 si n néfatif ou nul
  if (n<=0) return O;
  if (n==1 || n==2) return n ;
  return  suite(n-1) + suite(n-2) - 5 ;
}

// exercice 4.

typedef struct liste{
   int val ;
   struct liste * suivant ;
} Liste ;

void parcours(Liste * l){
  if ( ! vide(liste)) { // si l pas vide
     if ((l->val) > 10) printf("%d ", l->val);
     l=l->suivant ;
  }
}

// exercice 5. fonction récursive qui réalise une recherche
// du dernier élément d'une liste

typedef struct liste{
   int val ;
   struct liste * suivant ;
} Liste ;

Liste * dernier(Liste * l){ // retourne NULL si vide
  if (vide(liste)) return NULL ;
  if (vide(l->suivant) ) return l ;
  return dernier(l->suivant) ;
}

// exrecice 6. une fonction récursive qui inverse une chaîne de caractères

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

void inverse(char * c){
char * pa,  b;
int l=strlen(c) ;
 if ( l<=1 ) return;
 b = c[0];
 strcpy(c,c+1);
 inverse (c);
 c[l-1]=b;
 return;
}

main(){
 char tab[20], * ch="hello";
 strcpy(tab,ch);
 printf("initial : %s \n",tab);
 inverse(tab) ;
 printf("inversé : %s \n",tab);
}
```