

Discipline de programmation

1 Code modulaire

Extrait de listechaine.c, qui implémente les primitives des listes chaînées :

```
typedef struct maillon {int valeur; struct maillon* suivant;} maillon;
typedef struct listechaine {maillon* premier;} listechaine;

listechaine* creer(){
    listechaine* l = malloc(sizeof(listechaine));
    l->premier = NULL;
    return l;}

bool est_vide(listechaine* l) {
    return l->premier == NULL;}

int tete(listechaine *l) {
    assert(l->premier!=NULL);
    return l->premier->valeur;}
```

Extrait de listechaine.h, header du fichier précédent :

```
typedef struct maillon {int valeur; struct maillon* suivant;} maillon;
typedef struct listechaine {maillon* premier;} listechaine;

listechaine* creer();

bool est_vide(listechaine* l);

int tete(listechaine *l);
```

Contenu de moncode.c, qui utilise la liste chaînée :

```
1  #include "listechaine.h"
2  #include <stdio.h>
3  #include <stdlib.h>
4  #include <assert.h>
5  #include <stdbool.h>
6
7  int longueur(listechaine* l){
8      if (est_vide(l)){return 0;}
9      else{return 1+longueur(queue(l));}
10 }
11
12 int main(){
13     listechaine* l = creer();
14     ajouter_en_tete(2,l);
15     ajouter_en_tete(4,l);
16     ajouter_en_tete(6,l);
17     ajouter_en_tete(1,l);
18     ajouter_en_tete(10,l);
19     printf("%d\n", longueur(l));
20 }
```

2 Inclusions multiples

Extrait de grandpere.h, qui définit mastruct :

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

struct mastruct{int champ;;}
```

Extrait de pere.h, qui définit superfonction :

```
#include "grandpere.h"
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>

int superfonction(struct mastruct s);
```

Contenu de enfant.c, qui utilise mastruct et superfonction :

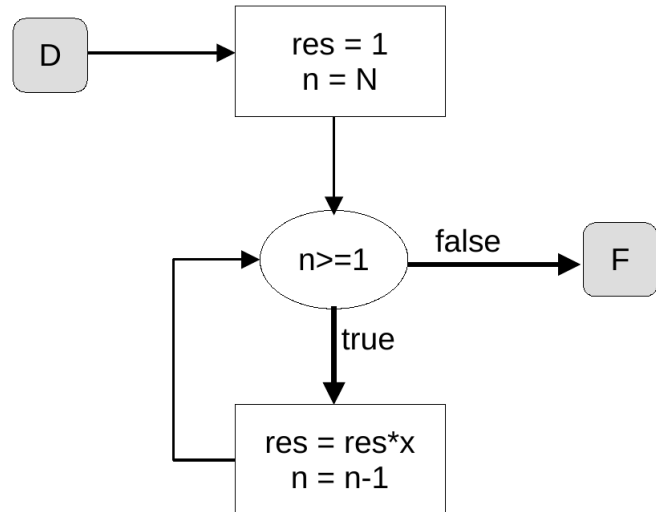
```
#include "grandpere.h"
#include "pere.h"
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>

int main(){
    struct mastruct s = {.champ=1};
    printf("%d\n",superfonction(s));
}
```

Graphes de flot de contrôle

Voici un algorithme et son graphe de flot de contrôle :

```
int puissance(int x, int N){
  int res = 1;
  int n = N;
  while(n>=1){
    res = res*x;
    n = n-1;
  }
}
```



Exemples de jeux de tests (tirés du livre Informatique MP2I/MPI de Vincent Barra)

