

Initiation à la programmation en C

Correction du TP n°9

Antoine Miné

26 avril 2007

Site du cours: <http://www.di.ens.fr/~mine/enseignement/prog2006/>

Exercice 1. Implantation des tableaux dynamiques.

```
----- tab.c -----
#include <stdio.h>
#include <stdlib.h>
#include <assert.h>

#include "tab.h"

struct tab* cree_tab()
{
    struct tab* t = malloc(sizeof(struct tab));
    assert(t);
    t->nb = 0;
    t->max = 20;
    t->mem = malloc(t->max);
    assert(t->mem);
    return t;
}

void detruit_tab(struct tab* t)
{
    free(t->mem);
    free(t);
}

void ajoute_char(struct tab* t, char c)
{
    if (t->nb>=t->max) {
        t->max *= 2;
        t->mem = realloc(t->mem, t->max);
        assert(t->mem);
    }
    t->mem[t->nb] = c;
    t->nb++;
}

void ajoute_string(struct tab* t, char* s)
{
    /* on pourrait faire plus efficace ici */
    for (; *s; s++) ajoute_char(t, *s);
}

void vide_tab(struct tab* t)
```

```

{
    t->nb = 0;
    /* on pourrait réduire la taille de mem ici */
}

void affiche_tab(struct tab* t)
{
    int i;
    for (i=0; i<t->nb; i++) fputc(t->mem[i], stdout);
    fputc('\n', stdout);
}

```

Exercice 3. Systèmes de Lindenmayer.

```

struct tab* subst(struct tab* arg)
{
    struct tab* res = cree_tab();
    int i;
    for (i=0; i<arg->nb; i++) {
        char c = arg->mem[i];
        if (c=='X') ajoute_string(res, "XgYag");
        else if (c=='Y') ajoute_string(res, "daXdY");
        else ajoute_char(res, c);
    }
    return res;
}

struct tab* calcule(int nb)
{
    int i;
    struct tab*t = cree_tab();
    ajoute_char(t, 'X');
    for (i=0; i<nb; i++) {
        struct tab*r = subst(t);
        detruit_tab(t);
        t = r;
    }
    return t;
}

```

Exercice 4. Courbe du Dragon.

```

void genere_postscript(struct tab* t)
{
    FILE* out = fopen("out.ps", "w");
    int i;
    if (!out) {
        printf("impossible de créer out.ps!\n");
        exit(1);
    }
    fprintf(out, "%!postscript\n"); /* en-tête post-script */
    fprintf(out, "297 419 moveto\n"); /* va au milieu de la page */
    for (i=0; i<t->nb; i++) {
        char c = t->mem[i];
        if (c=='d') fprintf(out, "90 rotate\n");
        else if (c=='g') fprintf(out, "-90 rotate\n");
    }
}

```

```
    else if (c=='a') fprintf(out, "1 0 rlineto\n");  
  
    }  
    fprintf(out, "stroke showpage\n"); /* affichage final */  
    fclose(out);  
}  
  
int main()  
{  
    genere_postscript( calcule(16) );  
    return 0;  
}
```

