

Correction code TD2

Exercice 1

3)

```
while(liste) {  
    nelt=liste->suivant;  
    free(liste);  
    liste=nelt;  
}
```

TD Ecosystème - Mise en place

Exercice 2

2)

```
toto@ordi:~/:gcc -Wall -c ecosys.c  
toto@ordi:~/:gcc -Wall -c main_test.c  
toto@ordi:~/:gcc -Wall -c main_ecosys.c  
toto@ordi:~/:gcc -Wall -o test_ecosys ecosys.o main_test.o  
toto@ordi:~/:gcc -Wall -o ecosys ecosys.o main_ecosys.o
```

3)

```
toto@ordi:~/:gcc -Wall -c main_test.c  
toto@ordi:~/:gcc -Wall -o test_ecosys ecosys.o main_test.o
```

4)

```
toto@ordi:~/:gcc -Wall -c ecosys.c  
toto@ordi:~/:gcc -Wall -o test_ecosys ecosys.o main_test.o  
toto@ordi:~/:gcc -Wall -o ecosys ecosys.o main_ecosys.o
```

5)

```
CFLAGS = -g -Wall -Wextra -pedantic  
CC = gcc
```

```
#PROGRAMS = tests_ecosys ecosys ecosys_param  
PROGRAMS = tests_ecosys ecosys
```

```
.PHONY: all clean  
.SUFFIX:
```

```
all: $(PROGRAMS)
```

```
tests_ecosys: ecosys.o main_tests.o  
    $(CC) -o $@ $(CFLAGS) $^
```

```
ecosys_param: ecosys.o main_param.o  
    $(CC) -o $@ $(CFLAGS) $^
```

```
ecosys: ecosys.o main_ecosys.o  
    $(CC) -o $@ $(CFLAGS) $^
```

```
#main_tests.o: main_tests.c  
# gcc -c $(CFLAGS) main_tests.c
```

```
#main_ecosys.o: main_ecosys.c
```

```
# gcc -c $(CFLAGS) main_ecosys.c
#Ou plus simplement
%.o:%.c %.h
$(CC) $(GCC_FLAGS) -c $<
clean:
rm -f *.o *~ $(PROGRAMS)
```

Exercice 3

```
1)
typedef struct _animal {
    int x;
    int y;
    int dir[2]; /* direction courante sous la forme (dx, dy) */
    float energie;
    struct _animal *suivant;
} Animal;

2)
Animal *creer_animal(int x, int y, float energie) {
    Animal *na = (Animal *)malloc(sizeof(Animal));
    assert(na);
    na->x = x;
    na->y = y;
    na->energie = energie;
    na->dir[0] = rand() % 3 - 1;
    na->dir[1] = rand() % 3 - 1;
    na->suivant = NULL;
    return na;
}

3)
Animal *ajouter_en_tete_animal(Animal *liste, Animal *animal) {
    assert(animal);
    assert(!animal->suivant);
    animal->suivant = liste;
    return animal;
}

4)
unsigned int compte_animal_rec(Animal *la) {
    if (!la) return 0;
    return 1 + compte_animal_rec(la->suivant);
}
/*Fourni*/
unsigned int compte_animal_it(Animal *la) {
    int cpt=0;
    while (la) {
        ++cpt;
        la = la->suivant;
    }
}
```

```

    return cpt;
}

```

Exercise 4

```

void afficher_ecosys(Animal *liste_proie, Animal *liste_predateur) {
    unsigned int i, j;
    char ecosys[SIZE_X][SIZE_Y];
    int nbpred=0,nbproie=0;
    Animal *pa=NULL;

    /* on initialise le tableau */
    for (i = 0; i < SIZE_X; ++i) {
        for (j = 0; j < SIZE_Y; ++j) {
            ecosys[i][j]=' ';
        }
    }
    /* on ajoute les proies */
    pa = liste_proie;
    while (pa) {
        ++nbproie;
        assert (pa->x >= 0 && pa->x < SIZE_X && pa->y >= 0 && pa->y < SIZE_Y);
        ecosys[pa->x][pa->y] = '*';
        pa=pa->suivant;
    }

    /* on ajoute les predateurs */
    pa = liste_predateur;
    while (pa) {
        ++nbpred;
        assert (pa->x >= 0 && pa->x < SIZE_X && pa->y >= 0 && pa->y <
            SIZE_Y);
        if ((ecosys[pa->x][pa->y] == '@') || (ecosys[pa->x][pa->y] == '*')) { /*
            proies aussi present */
            ecosys[pa->x][pa->y] = '@';
        } else {
            ecosys[pa->x][pa->y] = 'O';
        }
        pa = pa->suivant;
    }

    /* on affiche le tableau */
    printf("+");
    for (j = 0; j < SIZE_Y; ++j) {
        printf("-");
    }
    printf("+\n");
    for (i = 0; i < SIZE_X; ++i) {
        printf("|");
        for (j = 0; j < SIZE_Y; ++j) {
            putchar(ecosys[i][j]);
        }
    }
}

```

```
        printf("\n");
    }
    printf("+");
    for (j = 0; j<SIZE_Y; ++j) {
        printf("-");
    }
    printf("+\n");
    printf("Nb proies : %5d\tNb predateurs : %5d\n", nbproie, nbpred);
}
```