Programmation Impérative 2 - L2 Informatique

Solutions exo TD 4

Exercice I -

```
#include <stdlib.h>

struct fish {
   const char *name;
   const char *species;
   int teeth;
   int age;
};

void catalog(struct fish f) {
   printf("%s is a %s with %d teeth. He is %d\n", f.name , f.species , f.teeth , f.age );
}

int main() {
   struct fish snappy = {"Snappy", "Piranha", 69, 4};
   catalog(snappy);

   return EXIT_SUCCESS;
}
```

Exercice II -

```
1 b
2 d
3 f
4 g
5 a
6 c
7 e
```

Exercice III -

1. ```
   struct fish snappy = {"Snappy", "Piranha", 69, 4, {{"meat",0.2},{"swim
   in the jacuzzi", 7.5}}};
   ```

2. ```
   void label(struct fish a)
   {
      printf("Name:%s\nSpecies:%s\n%d years old, %d teeth\n", a.name, a.species,
              a.age,a.teeth);
      printf("Feed with %.2f lbs of %s and allow to %s for %.2f hours\n",
              a.care.food.weight, a.care.food.ingredients,
              a.care.exercise.description, a.care.exercise.duration);
   }
   ```

Exercice IV -

```
1  b
2  j
3  a
4  d
5  e
6  g
7  c
8  f
9  i
10 h
11 k
12 l
```

Exercice V -

```
void display(island *start) {
  island *i = start;

  for (; i !=   NULL ; i = i->next) {
    printf("Name: %s open: %s-%s\n", i->name , i->opens, i-> closes);
  }
}
```

Exercice VI -

```
island* create(char *name) {
  island *i = (island*) allocation_mem(1,sizeof(island));
  i->name = (char *) allocation_mem(strlen(name)+1,sizeof(char));
  strcpy(i->name,name);
  i->opens = "09:00";
  i->closes = "17:00";
  i->next = NULL;
  return i;
}
```

A noter, la commande `i->name = strdup(name);` peut remplacer les 2 lignes au dessus de `i->name : (char *) ...;  strcpy(i->name,name);`

```
void release(island *start) {
  island *i = start;
  island *next = NULL;
  for (; i != NULL; i = next) {
     next = i->next ;
     libere_mem_peda((void **)&(i->name)); //cf TD3
     libere_mem_peda((void **)&i); // cf TD3
  }
}
```

```
void main(){
  island *start = NULL;
  island *i = NULL;
  island *next = NULL;
  char name[80];

  for(; fgets(name,80, stdin) != NULL ;i= next ){
    next = create(name);

    if (start == NULL)
       start =  next;
    if (i != NULL)
       i->  next = next;
  }
  display(start);
  release(start);
}
```

Exercice VII -

```
1 f
2 a
3 c
4 d
5 b
6 e
```