thougths.md

Thoughts on how to process the data from the input

Table of contens

Bash Input

```
$> cat anthill
3                                 # <=> nb ants
                                  #_
## start                          # | <=> starting room
0 1 0                             #_|
                                  #_
## end                            # | <=> ending room
1 3 0 #bedroom                    #_|
2 5 0                             # <=> room
# The next room is the kitchen    # <=> ignore
3 9 0
                                  #_
0-2                               #-|--> link between two rooms
2-3                               # |
3-2                               # | possible links
3-4                               #_|
```

Interesting sections

- Number of ants
- Starting zone
- Ending zone
- Rooms
- Links

Input

Type:

- > char **

Content:

- > file data

Output

Type:

- > char **

File Name:

- > stdout (1)

File Content:

```
#number_of_ants
3
#rooms
#start
FirstRoom 2 0
#end
LastRoom 9 0
OtherRooms 2 0
[...]
#tunnels
0-2
[...]
#moves
P1-2
[...]
```

Method

1. ***Locate*** the number of ants
2. ***Save*** the number of ants
3. ***Locate*** the **starting** room
4. ***Save*** the **starting** room
5. ***Locate*** the **end** room
6. ***Save*** the **end** room
7. ***While* rooms** still present:\
   i. Locate room
   ii. Save room
8. ***While* links** still present:\
   i. ***Locate* link**
   ii. ***Save* link**
9. ***Call* Solver**
10. ***Display* result**

Architecture

> {4. / 6. / 7.2. / 8.2.}: Store rooms

```
typedef struct rooms_s {
    char *name;
    int posx;
    int posy;
    bool is_start;
    bool is_end;
    int nb_occupents;
    const int nb_max_occupents;
} rooms_t;
```

> { 8. }: Find / store links:

```
enum types {
    STRING;
};
linked_list_t *ll = init_linked_list(0);
append(ll, "2-3", STRING);
```

or

```c
typedef struct tunnel_s {
    int start;
    int end;
    int weight;
} tunnel_t;
```