Lab 9

Data Structure

Lab 9 (due on the day before the next Lab Session)

1. Do p9.c

Evaluation criteria

Category	Evaluation	
P9	100	
Total	100	

- *Use GCC 4.8 version or GCC 5.4 version.*
- *No score will be given if the gcc version is different.*

Lab 9 Maze

- You should finish p9 (init, union, find, createMaze, printMaze, freeMaze) before the next lab session and submit it to git.
- Folder name : Lab9
- code name: p9
- -15 score , if the folder, code names are wrong.
- -5 per code, if it does not use FILE I/O
- Each code will be tested by 5 different input files.
- 20 score for each input, if you don’t get the answer you get 0 score.

Lab 9 Maze

void init(DisjointSets *sets, DisjointSets *maze_print, int num) Initialize all cells to sets and maze_print.

void union(DisjointSets *sets, int i, int j) Union two sets.

int find(DisjointSets *sets, int i) Find the set including the number and return the representative member of the set.

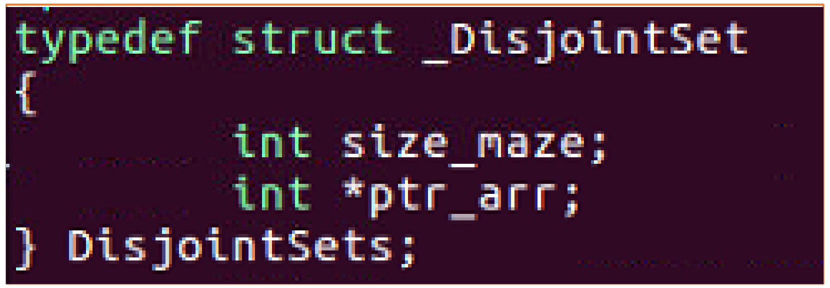
void createMaze(DisjointSets *sets, DisjointSets *maze_print, int num) Generate a maze that includes a path from Start position to End position **WITHOUT** any cycles. You can generate such a maze by randomly choosing a cell and direction. Use Union-Find ADT. For random number generation, use the library functions.

void printMaze(DisjointSets *sets, int num) Print the resulting maze.

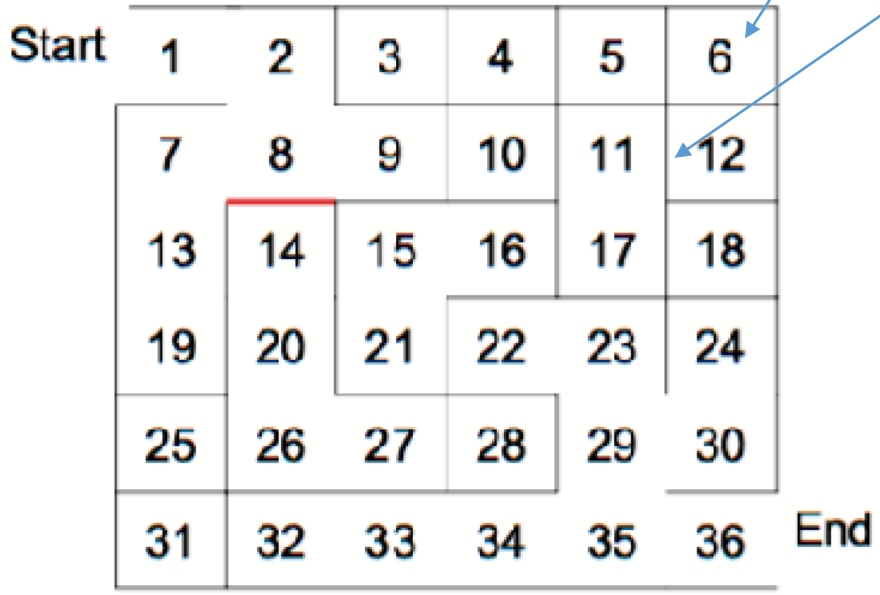
void freeMaze(DisjointSets *sets, DisjointSets *maze_print) Free memory of the maze.

Lab 9 Maze

- Structure

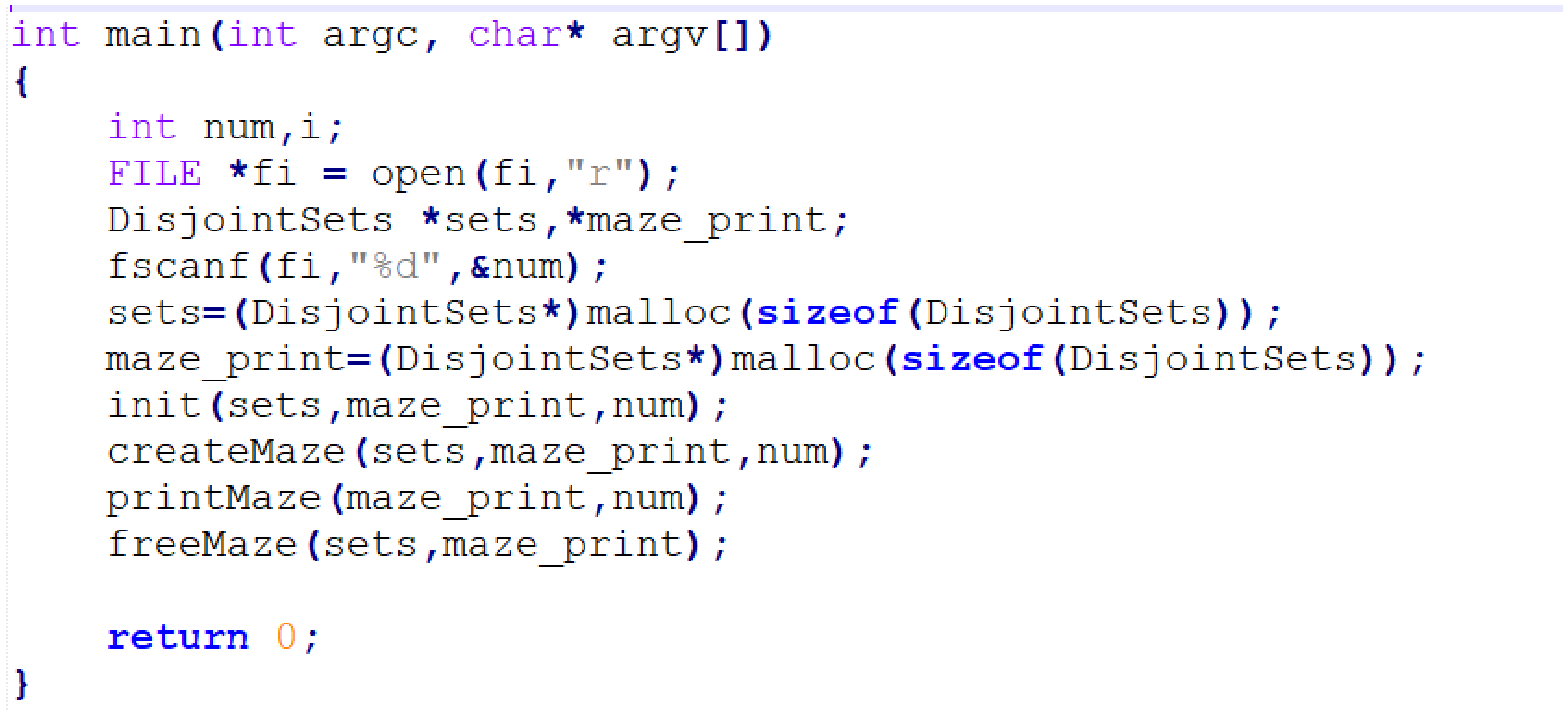


```
typedef struct _DisjointSet
{
        int size_maze;
        int *ptr_arr;
} DisjointSets;
```



- Variable
  - sets : means the number between the walls
  - maze_print : means the wall -1:yes,0:no

Lab 9 Maze



```c
int main(int argc, char* argv[])
{
	int num,i;
	FILE *fi = open(fi,"r");
	DisjointSets *sets,*maze_print;
	fscanf(fi,"%d",&num);
	sets=(DisjointSets*)malloc(sizeof(DisjointSets));
	maze_print=(DisjointSets*)malloc(sizeof(DisjointSets));
	init(sets,maze_print,num);
	createMaze(sets,maze_print,num);
	printMaze(maze_print,num);
	freeMaze(sets,maze_print);

	return 0;
}
```

Lab 9 Maze – Random Number Generation

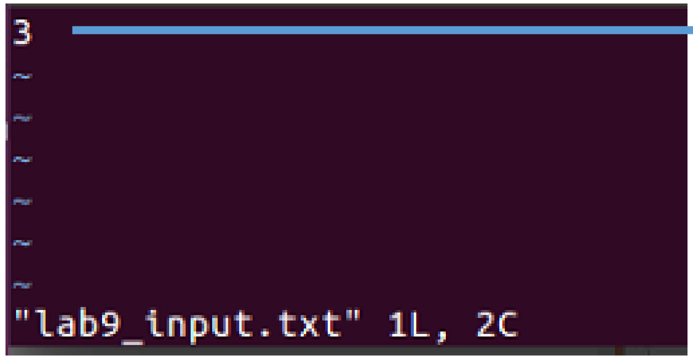
- Use srand() and rand() fuctions in <stdlib.h> and time() function in <time.h>

  exmaple

```
#include <stdlib.h>

#include <time.h>

srand((unsigned int)time(NULL)); // generate seed

…

int x = rand() // rand() function returns integer from 0 to 32767

…

int y = rand()%10 // y is from 0 to 9
```

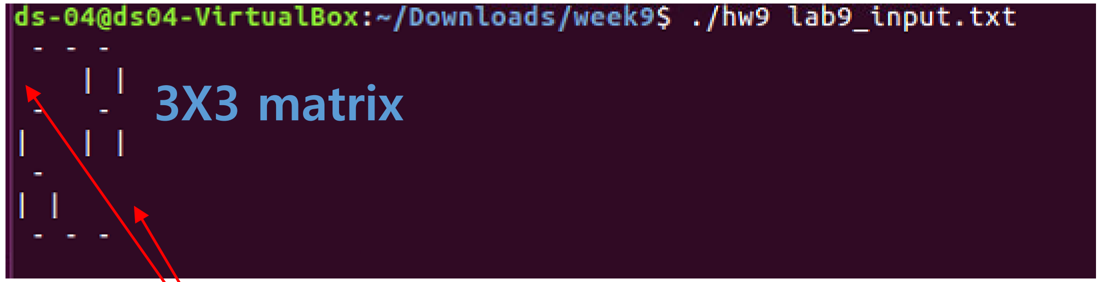
Lab 9 Maze – Example 1

- input file : Lab9_input1.txt



```
3
~
~
~
~
~
~
"lab9_input.txt" 1L, 2C
```

- Result



```
ds-04@ds04-VirtualBox:~/Downloads/week9$ ./hw9 lab9_input.txt
 - - -
    | |
 -   -
|   | |
 -
| |
 - - -
```

If 4, 4X4 matrix
If 5, 5X5 matrix
…
If n, nXn matrix

Lab 9 Maze

- program name : p9.c
- input : an integer in a file.
- output : the corresponding result in the standard output.