|  |  |
| --- | --- |
| **Mandatory Case** |  |
| Cmandatory |
| **Periode Berlaku** Semester Ganjil 2017/2018  ***Valid on*** *Odd Year 2017/2018* | **Software Laboratory Center**  **Assistant Recruitment 18-1** |

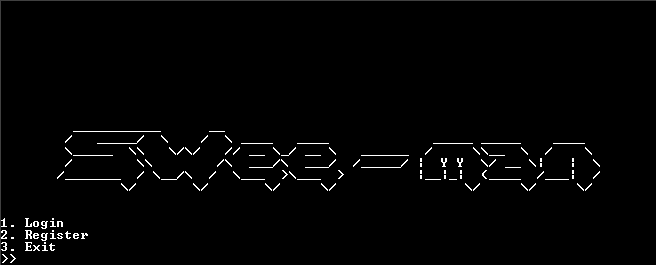
## Soal

*Case*

**SWee-Man**

**SWee-Man** is a new console game where you have to move **SWee-Man** to collect many stars. You as a programer asked to create this game using **C Programing Language** with this criteria.

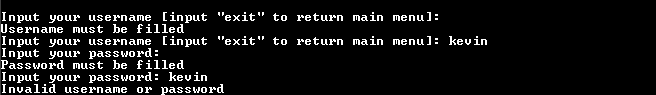
* At first, program will show **splash screen** and **main menu** then **read all user from user.txt**.



* The program consists of **3 menus**:

1. Login
2. Register
3. Exit

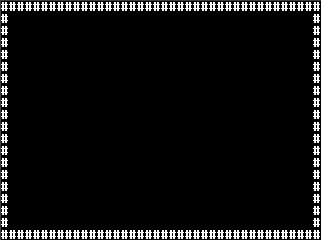
* If the user chooses **menu 1 “Login”**, then:
* Program will ask user to **input**:
* **Username**. Validate username **must be filled**.
* **Password**. Validate password **must be filled**.
* If all components are inputted, **check the username and password**. If **username and password correct**, show **welcome message** and **go to game menu**.
* Otherwise, **ask user to input username again**.
* If user **input username** **‘exit’** then **go to main menu**.



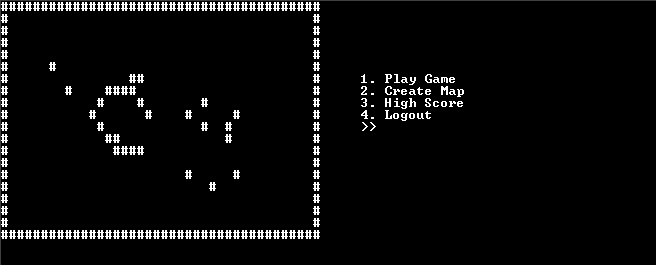
S:\01.AST\01.Case Making\04.NAR\18-1\CBP\SS\login2.PNG

* Game menu
* **Read all high score from score.txt** and **the** **map that user create**.
* Display **current active map in left side**. If user **has** **not create map yet**, there will **show default map**. Otherwise, **show map that user create**.
* Game menu consists of **4 menus**:

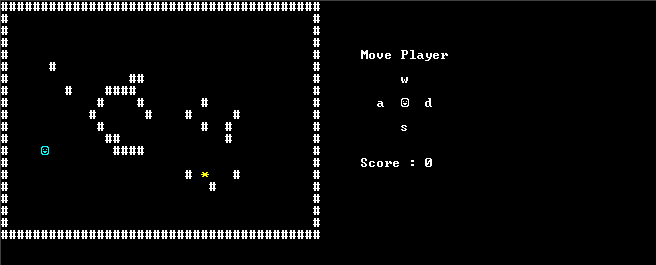
1. Play Game
2. Create Map
3. High Score
4. Logout

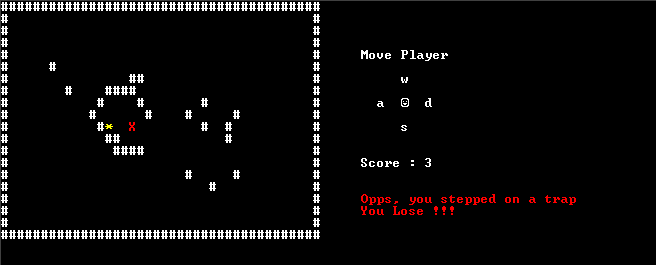


**default map 40 x 20**



* If the user chooses **menu 1 “Play Game”**, then:
* The game will **use your current active map**.
* Generate **star position**. Validate star position must be **inside the map** and **can’t same position with player, traps and wall**.
* Generate **player position**. Validate player position must be **inside the map** and **can’t same position with star, traps and wall**.
* Generate **ten traps position**. Validate trap position must be **inside the map** and **can’t same position with star, other trap and wall**.
* Usercan **control** the **SWee-man** by pressing **w** to **go up**, **a** to **go left, s** to **go down,** and **d** to **go right**. **Validate** that **SWee-man can’t move through a block**.
* If **player reach to star position** then will **increase score** and **generate new star position**.
* If **player reach to trap position**, then **change player position to ‘X’ symbol** and the game ends also show message “**Opps, you stepped on a trap. You Lose !!!**”
* Save the **last score to score.txt** and validate **only save top ten scores**.
* Program will wait until user **press enter button then program will return to game menu**.





* If the user chooses **menu 2 “Create Map”**, then:
* First **show your last currect active map**.
* Use **arrow to move cursor position**. Validate cursor position must **inside the map**.
* If user **press space button** at empty place there will **add the wall** and **decrease total block**.
* If user **press space button** at wall there will **remove the wall** and **increase total block**.
* Validate user **can only add 50 block wall**.
* If user **press esc button** validate the current design map **using flood fill algorithm**.
* If success then will **save the current design map to txt file** with following format:

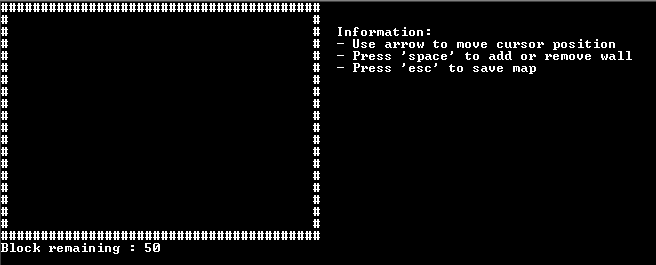
**[username]\_map.txt**

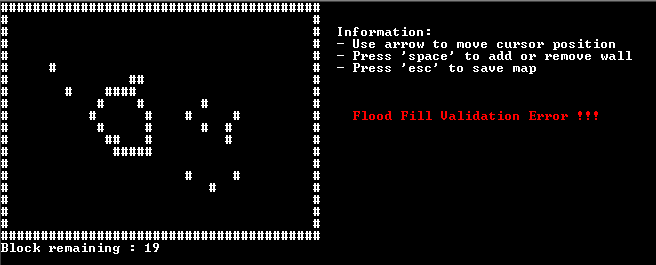
**Example:**

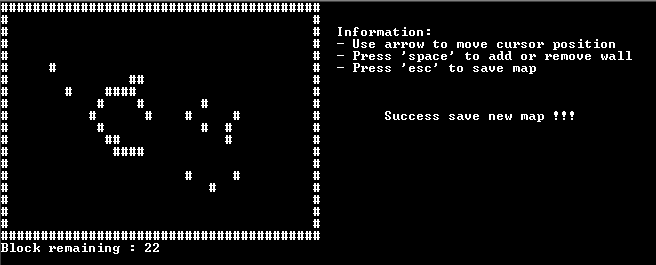
**username : kevinsudut**

**map file : kevinsudut\_map.txt**

* Otherwise, **keep user to design the map**.
* If user already create map then **overwrite last current map with new current map**.

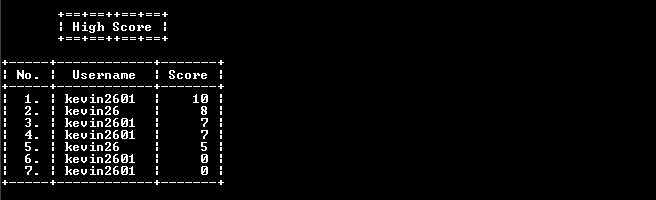




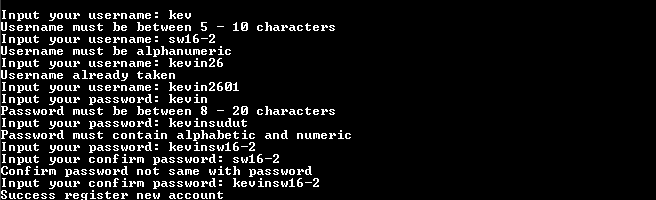


* If user choose **menu 3 “High Score”**, then:
* If there isn’t any high score, then program will show message “**No score available**”.
* Otherwise, show all score. The score will be **sorting with the biggest score is in the first line**.

S:\01.AST\01.Case Making\04.NAR\18-1\CBP\SS\score1.PNG



* If user choose **menu 4 “Logout”**, then return to main menu.
* If user choose **menu 2 “Register”**, then:
* Program will ask user to **input**:
* **Username**. Validate username length must be **between 5 – 10 characters**, must be **only contain alphabetic and numeric** and must be **unique (case insensitive)**.
* **Password**. Validate password length must be **between 8 – 20 characters** and must be **contain alphabetic and numeric**.
* **Confirm password**. Validate confirm password must be **same with password (case sensitive)**.
* If all components are inputted, **save new user to** **user.txt** and then return to main menu.



* If user choose **menu 3 “Exit”**, then:
* Program will **show splash screen** and **closed**.



* Additional information that you can use to create this program:

void gotoxy(int x, int y){

COORD c = {x, y};

SetConsoleCursorPosition(GetStdHandle(STD\_OUTPUT\_HANDLE), c);

}

* This function useful for move console cursor coordinate.
* The ( int x ) parameter is x coordinate of console and the ( int y ) parameter is y coordinate of console.
* Example use gotoxy function:

#include <stdio.h>

#include <Windows.h>

void gotoxy(int x, int y){

COORD c = {x, y};

SetConsoleCursorPosition(GetStdHandle(STD\_OUTPUT\_HANDLE), c);

}

int main(){

gotoxy(5, 3); //move cursor coordinate to x = 5 and y = 3

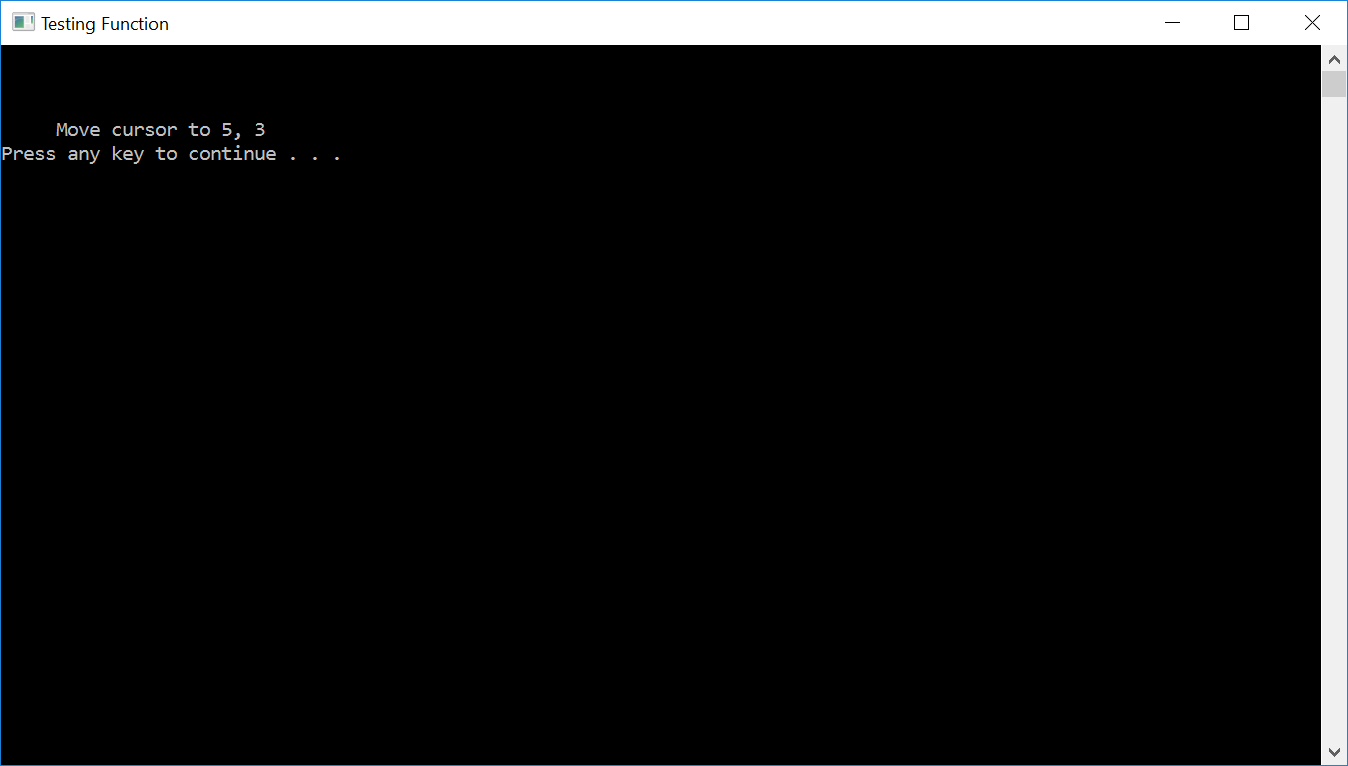
puts("Move cursor to 5, 3");

system("pause");

return EXIT\_SUCCESS;

}

**Output:**



void setColor(WORD w){

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), w);

}

* This function useful for change background and text color in console.
* The ( WORD w ) parameter accept the hex number.
* Example use setColor function:

#include <stdio.h>

#include <Windows.h>

void setColor(WORD w){

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), w);

}

int main(){

setColor(0x0F); //set console background to black and text to white

puts("0x0F");

setColor(0x0B); //set console background to black and text to light blue

puts("0x0B");

setColor(0x0E); //set console background to black and text to yellow

puts("0x0E");

setColor(0x0C); //set console background to black and text to red

puts("0x0C");

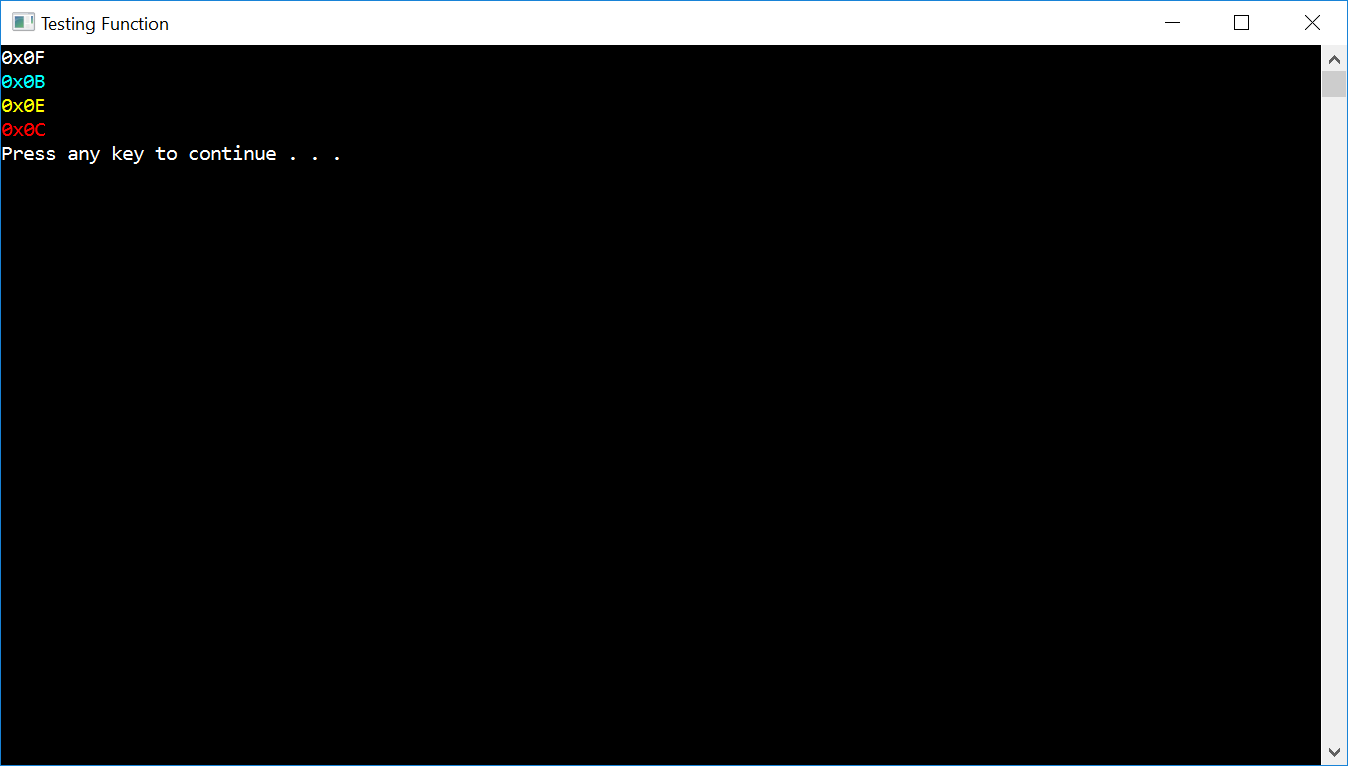
setColor(0x0F); //set console background to black and text to white

system("pause");

return EXIT\_SUCCESS;

}

**Output:**



* To request input for **move player** at **play game menu** and **cursor position** at **create map menu** use function from **conio.h (getch())**.

**Please run the EXE file to see the sample program.**