BBM 104 ASSIGNMENT-1 REPORT

➢ What is the problem and main purpose of this assignment

In this experiment, we are expected to build a game which is runned with commands from an input file and write expected output in a output file. The problem with this homework is read to the necessary files and to use these information appropriately.We have to take the information in these input files correctly and split them correctly.İt is really important for using these information as we desire. We have two input files. First one include informations of monsters and heroes. Second one include command lines. We have to obtain monsters and heroes information from first file and we need to keep this information on a sequential basis for every monster and hero because we will use this information later when they are necessary.Then we need to get the commands from the second file. İt will be better for us to get these commands line by line. By the way we will have to create a map to work on it. According to commands our map will be shaped. Each command can influence the shape of our map at the same time each command can affecting the properties of hero and monsters. By the way our output file will be created by these commands. When these commands are finished our file should be as required. The game can end in two ways. First, if all monsters and heroes die, game ends. Secondly, if all the commands in the file are ended, the game ends. By the way the output file should be appropriate when the game is over. The main purpose of the assignment is to manipulate the files correctly, and use the sequential information in the buffer correctly and to use the memory efficiently.

➢ solution in detail

First of all,I created a structures to keep the monsters and heroes specialties.I created a structure pointer from the type of these structures. Because we need to consider each hero and the monster separately. We need to dynamically allocate the necessary space for each hero and each monster. I obtained the monster and hero properties from the first file and sent them to arrays elements. Meanwhile I learned the number of heroes and monsters.I opened the second file at the beginning of the assignment to read the commands. when I assign the information in the second file to the variable of the command structure, I want it to continue until eof. I learned the line and column of the map from the "load map" command and created the map according to these values. I also obtain the map space dynamically. I created a structure for commands and I created an array from this structure and I sent every word in this array from second input file. In order to fulfill the commands, while reading the second file I compared the commands first words and second words then if it is true, I complete the necessary actions for each command.I took the coordinates of the characters in the put command and sent the first letters to the map.

There are 3 critical situations in the move command. If the first goes outside the bounds of the map, the character can not be moved. I said that if the desired coordinates are greater than the row and column variables of the map, I checked and prevented them from moving. The second can not move

the character if the desired coordinate is not empty. I said if the desired coordinates of the map were not the point, I checked and controlled the movement. Thirdly, if the character is dead, it can not be moved. I checked it if the characters HP was zero, and blocked the movement. If none of these situations existed, I made the move. I placed the characters in their new location on the map and put a dot in their old location.
When we came to attack command, we found the difference of the x axis of the characters. Then we found the difference of the y axis.then I get the power of these two and sum up them.this is our distance variable. if the distance variant is smaller than 2, it means that these two characters are neighbors. They can attack each other.If the character's hp is less than or equal to 0 then the character is dead.So ı made the characters hp and damage 0 in order to make the character ineffective. if the hero kills a monster, I increase xp of the hero. In the show command, I print all cases to output file with fprintf. Finally I closed the files and freeed the memory.

- **My Data Structure**

I created the Chars structure to hold the information of the characters. in this Chars structure I have the type, name, x axis, y axis, HP and damage of the characters. These features are different for each character. Each character has its own characteristic. so I created a pointer to this type of structure. Then I dynamically allocated space from the memory and I marked this first address in this pointer. then I placed each character in the order of this memory. So each character has its own characteristic and placed in memory in a row. This will make it easier for us to use them later. I also created a structure called COMMAND for the commands. and I added an array feature to keep each word in the second file separate. I created a pointer to this type of structure and dynamically allocated space for each commands. For the map, I took the row and column information from the second file and assigned two variables and using these variables dynamically allocate memory space. This field was taken in the size of these variables. I created 2d arrays with these variables and this was our map.

- **Explanation of my code**

I created CHARS structure for the keep characters information and created COMMAND structure for keep commands. I open the files with fopen because the first three arguments will be files. I have defined a structure pointer because each characters property will be different then I dynamically allocated space to hold every character and sent the first address of this place to the structers pointers. if the field is insufficient, I expanded the field with the expand function After that ı took characters proporties from first file and sent to pointers orderly.ı sent every proporties to characters with strtok function. Strtok function can splits strings.then I defined a pointer to type of struct COMMAND to hold each word separately. I dynamically allocated space to hold every commands word by word. Then ı started a loop that would read the words until the end of the file. I took the row and column values from the loadmap command on the first line and sent them to two int variable. then use these values to dynamically allocate space in RAM. Then ı created the map by using this spaces. Then I compare the first line of each command with the strcmp function and I do

necessary operations if it is correct. When the commands are finished, I filled in the output file with the show command. I increased the dead variable for each dead character in the attack function if this variable equals the number of characters I finished the game. or the commands are over, I have finished the game again. finally I closed the files and emptied the space which is allocated dynamically.

- **Tolga Furkan Güler**
- **21692874**