C Programming Language

#Project – 4

The Pokémon Tournament Game

Professor Oak is very grateful to you because his machine works again like a charm. He wants to give you a few of Pokémon to convey his thanks. He has shared a poster about upcoming Pokémon tournament of the town. The rules of the tournament are explained on the poster as follows:

- A player can participate in the tournament with at most four Pokémons.
- The tournament will take place on a space with 8x8 blocks (the area). Players can place their Pokémons on the first two closest rows to their locations.
- At each turn, the trainer can command only one of his/her Pokémons.
- If stamina of a Pokémon has been drained, the Pokémon must be removed from the area.
- The battle is over when a trainer has no Pokémon in the area.

Professor knows that you do not have a lot of information about Pokémons. Therefore, he has sent you a Pokedex. Pokedex is a machine that holds all the information about Pokémons. Professor told that there is only one Pokedex and he can only share it for a short while. He also tells you that you can reproduce the machine by yourself. We want to you become a participant of the tournament but first you should reproduce the Pokedex and take your gifted Pokémons from the professor. After that, you can be a magnificent Pokémon trainer and you can show your strategies at the tournament.

Part 1. Write a function that takes name, type, attack range, attack power and stamina specifications of the all Pokémons built a Pokedex. The function takes the name of the Pokémon that is queried by the user (inside the function) and show the specifications of the Pokémon if it is in the Pokedex database. Pokémons and attack types of Pokémons must be an enumerated type. You can find professor Oak's Pokedex information in Pokedex.xls file (you should not try to read information of the Pokémons from the file programmatically, just read the file to prepare your arrays and enums manually). The function prototype is:

```
void pokedex(char Pokemon_name[10][11], int range[], attack_type
            type[], int attack_power[], int stamina[])
```

Example run on the function:

Input:

```
Please type name of the Pokémon (type exit to close Pokedex):
Pikachu
```

Output:

```
Name   : Pikachu
A. Type : Linear
Range : 3 block
A. Power : 105
Stamina : 500
```

Part 2. Write a function that prints a menu as shown below:

Welcome to Laboratory of Professor Oak. How can I help you?

1) Show Pokémons
2) Catch a Pokémon
3) Release a Pokémon
4) Show my pocket
5) Back

This menu appears when the function is called and the user selects an option. The program will continuously ask options from the user until the user select the back option. The function prototype is:

```
void oaks_laboratory(char Pokemon_name[10][11], pokemon Pokemons[],
                     pokemon *my_Pokemons)
```

Write three functions to make this menu functional:

a. Show Pokémons option prints all the Pokémons (only the names of the Pokémons) to allow the user to see which Pokémons available to catch in the laboratory. Print the Pokémon names with their ids like the following output example. The prototype of required function for this option is:

```
void show_Pokemons(char Pokemon_name[10][11], pokemon
                   Pokemons[], int pokemon_count)
```

Output:

```
0 – Charmander
1 – Pikachu
2 – Squirtle
3 – Bulbasaur
4 – Pidgeotto
5 – Ratata
6 – Mug
7 – Caterpie
8 – Zubat
9 – Krabby
```

a. Catch a Pokémon option adds a Pokémon to the user's pocket. You should call the show_Pokémon () function to list all Pokémons before the user made his / her choice. Remember

that, the user can only select four different Pokémons: She / He can't catch a Pokémon more than once and can't catch more than four Pokémons. The prototype of required function for this option is:

```
void catch_a_pokemon(char Pokemon_name[10][11], pokemon Pokemons[], pokemon
                                        *my_pocket)
```

b. Release a Pokémon option removes a Pokémon that is already caught from the user's pocket. You should call the show_Pokemon() function to print the Pokémons that are in the user's pocket. The prototype of required function for this option is:

```
void release_a_pokemon(char Pokemon_name[10][11], pokemon Pokemons[],
                                        pokemon *my_pocket)
```

c. Show my pocket options also uses show_Pokemon () function to list user's Pokémon that are in his / her pocket.

Part 3. Write a function to simulate a single battle of the tournament. The function takes Pokémons of the user as an input and it allows user to locate his / her Pokémons on the fields which are allowed by the tournament rules (the user can locate his / her Pokémons on the first two row of the area, see the figure below). After the user has located the Pokémons, the battle starts with user's move. User picks one of his Pokémons to rule and she / he can move the selected Pokémon one or two block in all directions (except diagonal). After the user made his / her move, the function checks whether are there any Pokémon/s in the range of the ruled Pokémon's attack range. The ruled Pokémon attacks one or all Pokémons in its attack range according to its attack type. If attack type of the Pokémon is linear, than the Pokémon attacks a single Pokémon that is the closest to attacker (in attack range) in all directions (except diagonal). If there are more than one Pokémon which are stated equal distances, than attack all of them. On the other hand, if the attack type of the attacker Pokémon is quadratic, than attack all the Pokémon that are in the attack range of the Pokémon (the attacker can attack diagonal but don't consider Euclidian distance, just count the blocks as distance). After the Pokémon attack a rival Pokémon, the rival Pokémon's stamina decreases as much as attacker Pokémon's attack power and if the stamina of the rival Pokémon's is less or equal to zero than the rival Pokémon is removed from the area. After the user made his / her move, AI makes its move. The AI fights by using the rules following:

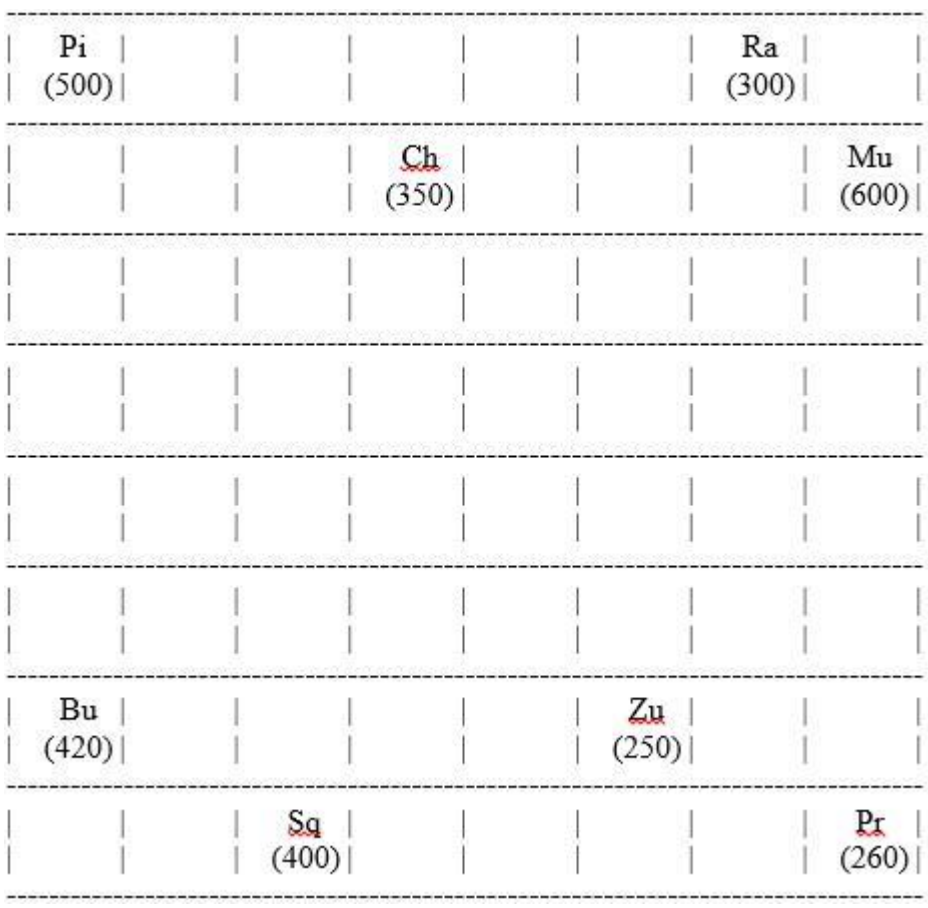
- The AI picks four Pokémon randomly at the beginning of the game and randomly locate the Pokémons on its permitted blocks (the AI can locate its Pokémons on the last two row of the area, see the figure below).
- After the user made his / her move, AI randomly select a Pokémon to rule. AI can move the selected Pokémon one or two block (randomly) in all directions (except diagonal), randomly.

Call the show_area () function after turn of a trainer (user or AI) is over. The function finishes the battle if one of the trainers (user or AI) doesn't have any Pokémon on the battle area. Check whether the user win the game or not by checking his / her pocket at the end of the battle and print the win situation (The user win the game or the user lost the game). The function prototype is:

```
void battle(char Pokemon_name[10][11], int range[], attack_type type[], int
          attack_power[], int stamina[], pokemon user_Pokemons[])
```

Part 4. Write a function that takes the battle area as an input and prints the current view of the arena as figure that is shown below:

(First row) (Don't print the line, just for clarification)



(Last row) (Don't print the line, just for clarification)

```
void show_area (char Pokemon_name[10][11],int area[8][8],int
int pokemon_staminas_view[8][8])
```

Part 5. Write a function that prints the menu which is shown below:

Please select an option to continue:

1) Open Pokedex
2) Go to Oak's Laboratory
3) Enter the tournament

The first option calls pokedex () function, the second one calls oaks_laboratory () function and the last option calls battle () function.

Notes:

- **For all functions consider all exceptions as possible as you can. In examples; the trainers can't violate the borders of the battle area, if the user wants to catch more than four Pokémon or he / she wants to release a Pokémon from an empty pocket, than show them a fail message and prevent the action etc.**
- **Don't use any extra library except the standard input/output library of the C language.**