```
/*Joud Talal Alharbi 2110207
 Shahd Alshamrani 2113082
 Noura Almutairi 20113615
 Raghad Hussain 2112736
 Raghad Basfar 2024092
 Joud Hattan  2010849
 */

package pokemongame;

public class PokemonLL {
    private PokemonLLnode head;

    //constructor
    public PokemonLL(){
    head=null;}

    //isEmpty
    public boolean isEmpty() {
        return head == null;
    }

    //print list
    public void printList() {

        PokemonLLnode help = head;

      while (help != null) {
            // Print the data value of the node
           System.out.println(help.getPokemon().ToString());
            // Step one node over
            help = help.getNext();
      }
    }

    //add
    public void addNode(Pokemon p){{
        //create new node
        PokemonLLnode a = new PokemonLLnode(p);
        // IF there is no list, newNode will be the first node, so just return it
        if (head == null ) {
            a.setNext(head);
            head = a;
      }
      // ELSE, we have a list. Insert the new node at the correct location
      else {
            // We need to traverse to the correct insertion location...so we need a
help ptr
            PokemonLLnode helpPtr = head;
            // Traverse to correct insertion point
            while (helpPtr.getNext() != null) {
            helpPtr = helpPtr.getNext();
            }
            // Now make the new node. Set its next to point to the successor node.
            // And then make the predecessor node point to the new node
            a.setNext(helpPtr.getNext());
            helpPtr.setNext(a);
      }
    }}
```

```
//•   The total number of Pokémons.
    public int NumOfPokemons(){
        while (!isEmpty()){
        int num=0;
        PokemonLLnode help=head;
        while(help!=null){
            num++;
        help=help.getNext();}
        return num;}

    return 0;}


//•   The total number of Pokémons for each type.
     public String NumOfPokemonType(){
          while (!isEmpty()){
          PokemonLLnode help=head;
          int water=0;
          int fire=0;
          int electric=0;
          int grass=0;

          while(help!=null){
          if (help.getPokemon().GetType().equals("Water"))
              water++;
          if (help.getPokemon().GetType().equals("Fire"))
              fire++;
          if(help.getPokemon().GetType().equals("Grass"))
              grass++;
          if(help.getPokemon().GetType().equals("Electric"))
              electric++;
          help=help.getNext();}

  return("Number of Electric pokemons = "+electric+"\n Number of Grass pokemons = 
"+grass+"\n Number of Fire pokemons = "+fire+"\n Number of Water pokemons = 
"+water);
     }
     return "empty list";}


//•   The strongest Pokémon in the list (the Pokémon with higher total power).
      public String StrongestPokemon(){
           while (!isEmpty()){
      PokemonLLnode help=head;
      Pokemon S=head.getPokemon();
      while(help!=null){
      if(help.getPokemon().GetTotal()>S.GetTotal())
          S=help.getPokemon();
      help=help.getNext();}

     return S.GetName(); }
      return null;}


//•   The average of all Pokémons power
       public double AvgPower(){
            while (!isEmpty()){
```

```
          double avg=0;
          PokemonLLnode help=head;
          while(help!=null){
              avg+=help.getPokemon().GetTotal();
      help=help.getNext();}
          return avg/NumOfPokemons();}
       return 0;}



     //----------search----------
    public Pokemon search(int id) {
        PokemonLLnode help = head;
      while (help != null) {
            if (help.getPokemon().GetId()==id) {
                return help.getPokemon();
            }
            help= help.getNext();
      }
      return null;
    }


}
```