**package** problem1;

**public** **class** MyStringList {

**private** **final** **int** INITIAL\_LENGTH = 4;

**private** String[] strArray;

**private** **int** size;

**public** **void** sort(){

**if**(**this**.strArray == **null** || **this**.strArray.length <=1) {

**return**;

}

**int** len = **this**.strArray.length;

**int** temp = 0;

**for**(**int** i = 0; i < len; ++i){

**int** nextMinPos = minpos(i,len-1);

swap(i,nextMinPos);

}

}

**void** swap(**int** i, **int** j)

{

String temp = **this**.strArray[i];

**this**.strArray[i] = **this**.strArray[j];

**this**.strArray[j] = temp;

}

//find minimum of arr between the indices bottom and top

**public** **int** minpos(**int** bottom, **int** top)

{

String m = **this**.strArray[bottom];

**int** index = bottom;

**for**(**int** i = bottom+1; i <= top; ++i)

{

**if** (**this**.strArray[i]!=**null**)

{

**if**(**this**.strArray[i].compareTo(m)<0)

{

m = **this**.strArray[i];

index = i;

}

}

}

//return location of min, not the min itself

**return** index;

}

**public** MyStringList() {

strArray = **new** String[INITIAL\_LENGTH];

size = 0;

}

**public** **void** add(String s){

**if**(size == strArray.length) resize();

strArray[size++] = s;

}

**public** String get(**int** i){

**if**(i < 0 || i >= size){

**return** **null**;

}

**return** strArray[i];

}

**public** **boolean** find(String s){

**for**(String test : **this**.strArray)

{

**if** (test != **null**)

{

//System.out.printf("%s -- %s -- %s \n",test,s,test.equals(s));

**if**(test.equals(s))

{

**return** **true**;

}

}

}

**return** **false**;

}

**public** **void** insert(String s, **int** pos){

**if**(pos > size) **return**;

**if**(pos >= strArray.length||size+1 > strArray.length) {

resize();

}

String[] temp = **new** String[strArray.length+1];

System.*arraycopy*(strArray,0,temp,0,pos);

temp[pos] = s;

System.*arraycopy*(strArray,pos,temp,pos+1, strArray.length - pos);

strArray = temp;

++size;

}

**public** **boolean** remove(String s){

**if**(size == 0) **return** **false**;

**int** index = -1;

**for**(**int** i = 0; i < size; ++i ){

**if**(strArray[i].equals(s)){

index = i;

**break**;

}

}

**if**(index==-1) **return** **false**;

String[] temp = **new** String[strArray.length];

System.*arraycopy*(strArray,0,temp,0,index);

System.*arraycopy*(strArray,index+1,temp,index,strArray.length-(index+1));

strArray = temp;

--size;

**return** **true**;

}

**private** **void** resize(){

System.***out***.println("resizing");

**int** len = strArray.length;

**int** newlen = 2\*len;

String[] temp = **new** String[newlen];

System.*arraycopy*(strArray,0,temp,0,len);

strArray = temp;

}

**boolean** recurse(**int** a, **int** b, String val) {

**int** mid = (a+b)/2;

**if**(**this**.strArray[mid].compareTo(val)==0)

{

**return** **true**;

}

**if**(a > b) {

**return** **false**;

}

**if**(**this**.strArray[mid].compareTo(val)>=1)

{

**return** recurse(mid+1, b, val);

}

**return** recurse(a,mid-1,val);

}

//search a sorted array

**boolean** search(String val) {

**boolean** b = recurse(0,**this**.strArray.length-1, val);

**return** b;

}

**public** String toString(){

StringBuilder sb = **new** StringBuilder("[");

**for**(**int** i = 0; i < size-1; ++i){

sb.append(strArray[i]+", ");

}

sb.append(strArray[size-1]+"]");

**return** sb.toString();

}

**public** **int** size() {

**return** size;

}

**public** **static** **void** main(String[] args){

String[] datainit= {"big", "small", "tall", "short", "round", "square",

"enormous", "tiny","gargantuan", "lilliputian",

"numberless", "none", "vast", "miniscule"};

MyStringList l = **new** MyStringList();

**for** (String string : datainit) {

l.add(string);

}

System.***out***.println("Solution point A");

System.***out***.println(l);

System.***out***.println("done");

System.***out***.println("Solution point B");

String[] databaseSearch= {"number","tiny"};

l.sort();

**for** (String string : databaseSearch) {

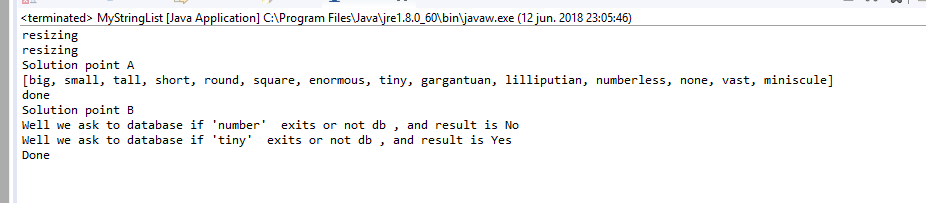
System.***out***.printf("Well we ask to database if '%s' exits or not db , and result is %s \n",string , (l.find(string) ? "Yes" : "No" ) );

}

System.***out***.println("Done");

}

}



Problem 2

**package** problem2;

**import** java.util.Arrays;

**public** **class** MyPersoList {

**private** **final** **int** INITIAL\_LENGTH = 4;

**private** Person[] strArray;

**private** **int** size;

**public** MyPersoList() {

strArray = **new** Person[INITIAL\_LENGTH];

size = 0;

}

// Add element in last

**public** **void** add(Person s){

**if**(size == strArray.length) resize();

strArray[size++] = s;

}

**public** Person get(**int** i){

**if**(i < 0 || i >= size){

**return** **null**;

}

**return** strArray[i];

}

**public** **boolean** find(String lastName)

{

**for**(Person test : strArray)

{

**if**(test.getLast().equals(lastName))

{

**return** **true**;

}

}

**return** **false**;

}

/\*public void insert(String s, int pos){

if(pos > size) return;

if(pos == strArray.length||size+1 > strArray.length) {

resize();

}

String[] temp = new String[strArray.length+1];

System.arraycopy(strArray,0,temp,0,pos);

temp[pos] = s;

System.arraycopy(strArray,pos,temp,pos+1, strArray.length - pos);

strArray = temp;

++size;

}\*/

**public** **void** insert(Person s, **int** pos)

{

**if**(pos<0 || pos > size)

{

**return**;

}

**if**(pos == strArray.length||size+1 > strArray.length)

{

**this**.resize();

}

Person[] temp = **new** Person[strArray.length+1];

**for**(**int** i = 0; i < pos; i++)

{

temp[i] = strArray[i];

}

temp[pos] = s;

**for**(**int** i = pos + 1; i < strArray.length; i++)

{

temp[i] =strArray[i - 1];

}

strArray = temp;

++size;

}

**public** **boolean** remove(Person s){

**if**(size == 0) {

**return** **false**;

}

**int** index = -1;

**for**(**int** i = 0; i < size; ++i )

{

**if**(strArray[i].equals(s)){

index = i;

**break**;

}

}

**if**(index==-1) {

**return** **false**;

}

Person[] temp = **new** Person[strArray.length];

System.*arraycopy*(strArray,0,temp,0,index);

System.*arraycopy*(strArray,index+1,temp,index,strArray.length-(index+1));

strArray = temp;

--size;

**return** **true**;

}

**private** **void** resize(){

System.***out***.println("resizing");

**int** len = strArray.length;

**int** newlen = 2\*len;

Person[] temp = **new** Person[newlen];

System.*arraycopy*(strArray,0,temp,0,len);

// strArray = Arrays.copyOf(strArray, newlen);

strArray = temp;

}

**public** String toString(){

StringBuilder sb = **new** StringBuilder("[");

**for**(**int** i = 0; i < size-1; ++i){

sb.append(strArray[i]+", ");

}

sb.append(strArray[size-1]+"]\n");

**return** sb.toString();

}

**public** **int** size() {

**return** size;

}

**public** **boolean** isEmpty(){

**return**(size==0);

}

**public** Object clone()

{

Person[] temp = Arrays.*copyOf*(strArray, size);

**return** temp;

}

**public** **static** **void** main(String[] args) {

MyPersoList l = **new** MyPersoList();

String last = "Mosquera";

String person\_to\_find = "Torrijos";

l.add(**new** Person(last, "Bob", 21));

l.add(**new** Person(last, "Steve", 13) );

l.add(**new** Person(last, "Susan", 19) );

l.add(**new** Person(last, "Mark", 43) );

l.insert(**new** Person("Torrijos", "Renuka", 11) , 4);

System.***out***.println(l);

l.add(**new** Person(last, "Dave", 15) );

System.***out***.println("The list of size "+l.size()+" is "+l);

l.remove(**new** Person(last, "Mark", 18) );

l.remove(**new** Person(last, "Bob", 28) );

System.***out***.println("The list of size "+l.size()+" is "+l);

l.insert(**new** Person(last, "Richard", 28) ,3);

System.***out***.println("The list of size "+l.size()+" after inserting Richard into pos 3 is "+l);

l.insert(**new** Person(last, "Tonya", 78) ,0);

System.***out***.println("The list of size "+l.size()+" after inserting Tonya into pos 0 is "+l);

Person[] x = (Person[]) l.clone();

System.***out***.println(Arrays.*toString*(x));

/// well we now to find a person using method find

System.***out***.printf("Anybody with Last name %s is on our database %s \n",person\_to\_find, ( l.find(person\_to\_find) ? "Yes" : "No") );

}

}

