**public** **class** Start {

/\*\*

\* **@param** args

\*/

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

MyArrayList my = **new** MyArrayList(2,2);

my.add(1000000);

my.add(7);

my.add(9);

my.add(123);

my.add(200);

my.add(2123);

my.add(26);

my.add(62);

my.add(11);

my.add(3);

my.add(7100);

//System.out.println("Start MyArray" + my);

//System.out.println("Number of change" + my.change(7, 11111));

//System.out.println("11111 is expected two times" + my);

//my.change(100, 2, 0);

//System.out.println("11111 is expected only one time" + my);

System.*out*.println(my);

my.selectionSort();

System.*out*.println(my);

}

}

**public** **class** MyArrayList {

**int** [] elements;

**int** growRate;

**int** pos;

**public** MyArrayList(**int** size, **int** growRate)

{

elements = **new** **int**[size];

**this**.growRate = growRate;

pos = 0;

}

**private** **void** grow()

{

**int** [] tempElements = **new** **int** [elements.length \* growRate];

**for** (**int** i = 0; i < elements.length; i++) {

tempElements[i] = elements[i];

}

elements = tempElements;

}

**public** **void** add(**int** newElement)

{

**if**(pos == elements.length)

grow();

elements[pos] = newElement;

pos++;

//System.out.println(pos);

}

**public** String toString()

{

String temp = "";

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

temp = temp + " " + elements[i];

}

**return** temp;

}

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

{

**return** pos;

}

**public** **int** removeLastElement()

{

**if**(pos == 0)

{

**return** Integer.*MAX\_VALUE*; //very bad code :)

}

**else**

{

pos--;

**return** elements[pos-1];

}

}

**public** **void** eraseAll()

{

pos = 0;

}

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

{

**return** pos == 0;

}

**public** **int** change(**int** find, **int** replace)

{

**int** numberOfChanged = 0;

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

**if**(elements[i] == find)

{

elements[i] = replace;

numberOfChanged++;

}

}

**return** numberOfChanged;

}

**public** **int** change(**int** maxChanges, **int** find, **int** replace)

{

**int** numberOfChanged = 0;

**for** (**int** i = 0; i < pos && maxChanges > numberOfChanged; i++)

{

**if**(elements[i] == find)

{

elements[i] = replace;

numberOfChanged++;

}

}

**return** numberOfChanged;

}

**public** **void** simpleSorting()

{

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

//System.out.println("After " + i + " external loop" + toString());

**for** (**int** j = i+1; j > 0 && elements[j]< elements[j-1]; j--) {

**int** temp = elements[j-1];

elements[j-1] = elements[j];

//System.out.println("Two equal numbers are expected" + toString());

elements[j] = temp;

//System.out.println("We dont expect two equal number" + toString());

}

}

}

**public** **int** sum()

{

**int** temp = 0;

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

temp = temp + elements[i];

}

**return** temp;

}

**public** MyArrayList compareSize(MyArrayList my)

{

**if**(**this**.pos < my.pos)

**return** my;

**else** **if**(**this**.pos > my.pos)

**return** **this**;

**else**

**return** **null**;

}

**public** MyArrayList compareSum(MyArrayList my)

{

**int** sumThis = sum();

**int** sumMy = my.sum();

**if**(sumThis > sumMy)

**return** **this**;

**else** **if**(sumThis < sumMy)

**return** my;

**else**

**return** **null**;

}

**public** **void** selectionSort()

{

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

**int** minUntilNow = i;

**for** (**int** j = i+1; j < pos; j++) {

**if**(elements[minUntilNow]> elements[j])

{

minUntilNow = j;

}

}

**int** temp = elements[i];

elements[i] = elements[minUntilNow];

elements[minUntilNow] = temp;

}

}

}

**public** **class** Start {

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

Person a = **new** Person("A", **true**);

Person b = **new** Person("B", **false**);

Person c = **new** Person("C", **false**);

Person d = **new** Person("D", **true**);

Person e = **new** Person("E", **false**);

Person f = **new** Person("F", **false**);

a.setChildren(c);

b.setChildren(c);

a.setLivingWith(b);

d.setLivingWith(c);

d.setChildren(e);

c.setChildren(e);

c.setChildren(f);

System.*out*.println(a);

}

}

**import** java.util.ArrayList;

**public** **class** Person {

**private** String name;

**private** **boolean** male;

**private** String placeOfBirth;

**private** String dateOfBirth;

**private** Person father;

**private** Person mother;

**private** Person livingWith;

**private** ArrayList <Person> children;

**private** ArrayList <Person> siblings;

**public** Person(String name, **boolean** male)

{

**this**.name = name;

**this**.male = male;

children = **new** ArrayList();

siblings = **new** ArrayList();

}

**public** Person(String name, String placeOfBirth, String dateOfBirth)

{

**this**.name = name;

**this**.placeOfBirth = placeOfBirth;

**this**.dateOfBirth = dateOfBirth;

children = **new** ArrayList();

siblings = **new** ArrayList();

}

**public** Person getFather() {

**return** father;

}

**public** **void** setFather(Person father) {

**this**.father = father;

}

**public** Person getMother() {

**return** mother;

}

**public** **void** setMother(Person mother) {

**this**.mother = mother;

}

**public** ArrayList getChildren() {

**return** children;

}

**public** **void** setChildren(Person p) {

children.add(p);

**if**(male)

p.setFather(**this**);

**else**

p.setMother(**this**);

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** ArrayList getSiblings() {

**return** siblings;

}

**public** **void** setSiblings(Person p) {

siblings.add(p);

}

**public** Person getLivingWith() {

**return** livingWith;

}

**public** **void** setLivingWith(Person livingWith) {

**if**(**this**.livingWith != livingWith)

{

**this**.livingWith = livingWith; //note that I am living with

livingWith.setLivingWith(**this**);

}

}

**public** String toString()

{

String temp = "";

**if**(!children.isEmpty())

{

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

temp = temp + " " + children.get(i);

}

}

**return** name + "sex" + male + "\n" + temp;

}

}