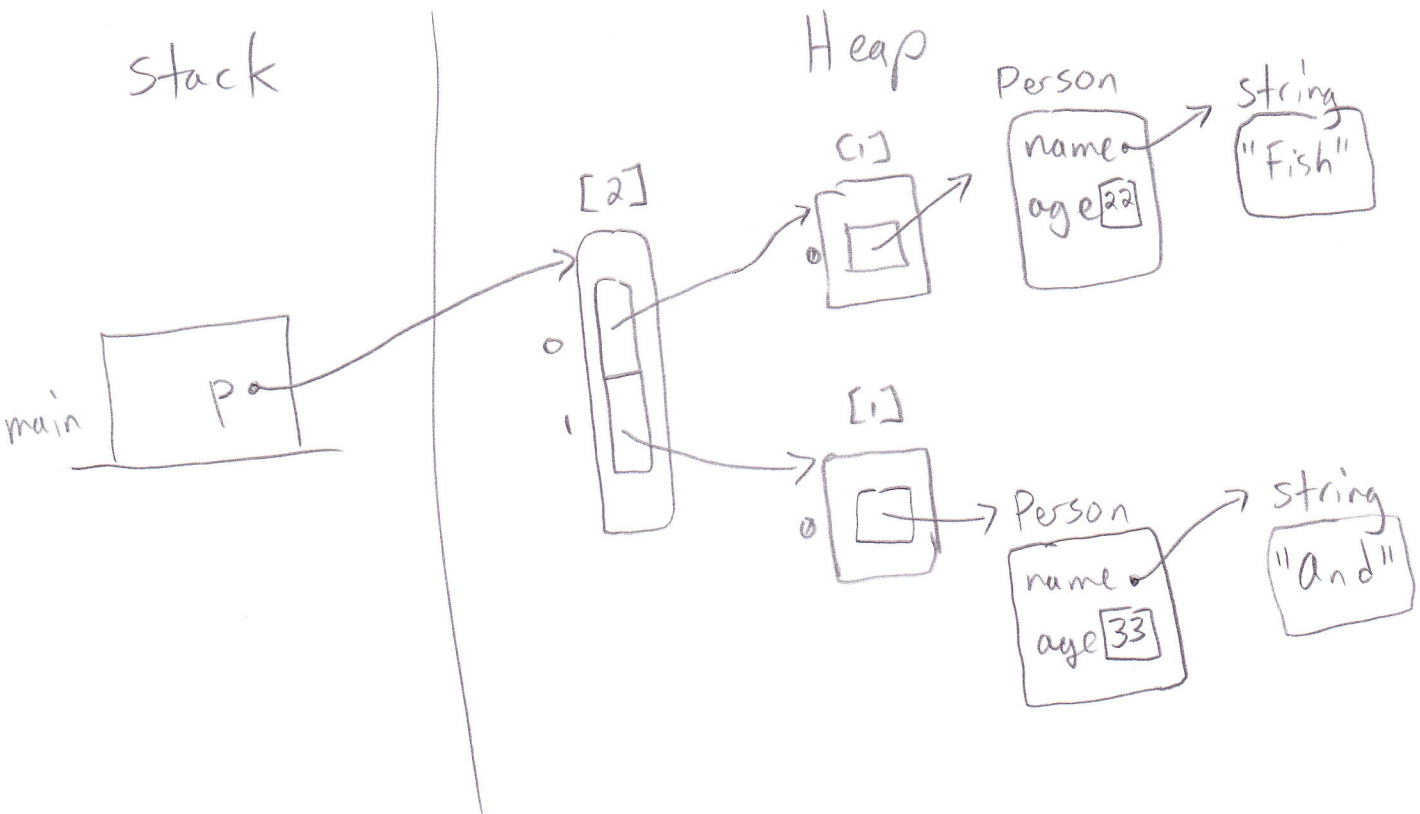


Part 2: Short Answer Questions 30 points [points shown for each question, spend time wisely]

1. Using the code provided draw a memory map to represent Stack and Heap just at the end of method main. [5 points]

```
public class Person {  
    private String name;  
    private int age;  
    public Person(String name, int age){ this.name = name; this.age = age;}  
    public Person(){ this("Person 1", 25); }  
    public String getName(){ return this.name; }  
    public void setName(String name){ this.name = name; }  
    public int getAge(){ return this.age; }  
    public void setAge(int age){ this.age = age; }  
}
```

```
public class PersonDemo{  
    public static void main(String[] args){  
        Person[][] p = new Person[2][1];  
        p[0][0] = new Person("Fish", 22);  
        p[1][0] = new Person("And", 33);  
        // draw memory map from here  
    }  
}
```

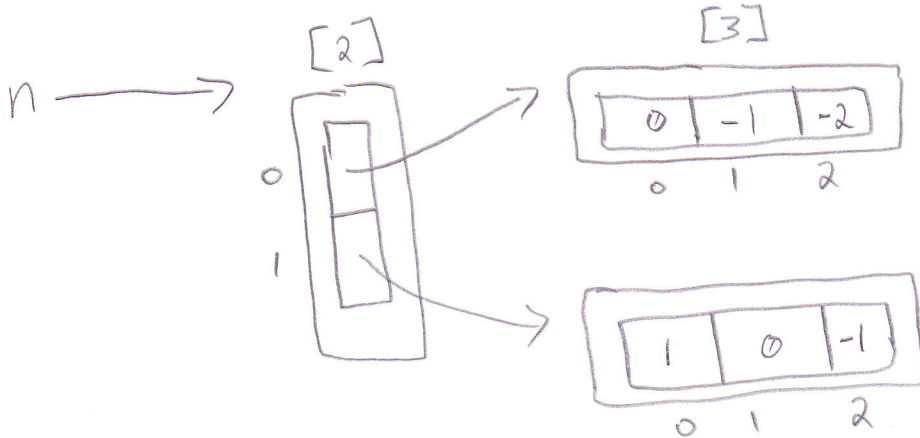


2. Write a loop so that the array shown will have its elements set to the value of subtracting the second index from the first one, e.g. element at index [0][1] will be set with a value of -1, element at index of [1][2] will be set to -1 (etc.)
[4 points]

```
int[][] n = new int[2][3];
```

```
for (int i=0; i < n.length; i++) {  
    for (int k=0; k < n[i].length; k++) {  
        n[i][k] = i - k;  
    }  
}
```

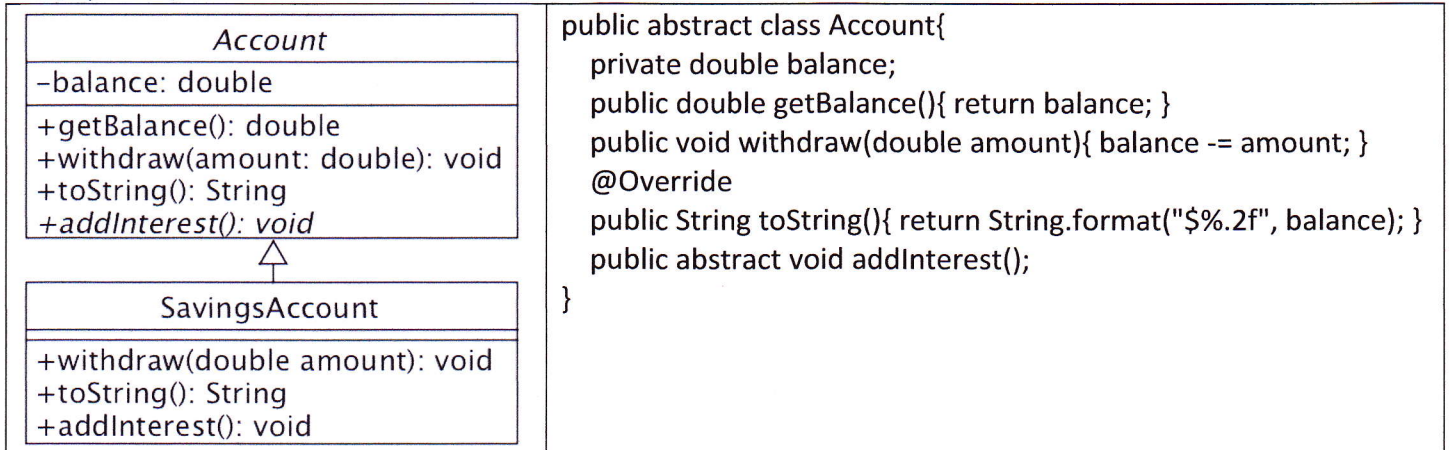
3. Draw a memory map of the array from question 2 after it has been initialized with the requested numbers.
[6 points]



4. Examine the figure below as well as the code for class Account then write the code for class SavingsAccount:

- SavingsAccount withdraw should deduct 10 cents from the withdrawal for a promotional event
E.g. withdraw 50 cents, but only 40 cents removed from balance
- Expected output from SavingsAccount toString is: Savings Account Balance: \$0.00
(Assuming the account was created but has a balance of zero)
- addInterest() should add 10% of the current balance to the balance as a negative withdrawal.
i.e. make the amount of interest into a negative number, then 'withdraw' it.

[5 points]



Tip: Don't forget extends, super., StringBuilder and @Override

```
public class SavingsAccount extends Account {
```

```
    @Override
```

```
    public void withdraw (double amount) {
        super.withdraw (amount - 0.10);
    }
```

```
}
```

```
    @Override
```

```
    public String toString () {
        StringBuilder b = new StringBuilder ();
```

```
        b.append ("Saving Account Balance :").append (super.toString());
        return b.toString ();
    }
```

```
}
```

```
    @Override
```

```
    public void addInterest () {
```

```
        double amount = getBalance () * 0.10 * -1 ;
        //super.withdraw or withdraw acceptable here.
        super.withdraw (amount);
    }
```

```
}
```

Test B

5. Using the words presented as a guide write a JUnit test method to test that methods getPrice() and setPrice(double) of class Pear work correctly. (Each word may be used more than once.)
[5 points]

Pear
-price: double
+getPrice(): double +setPrice(price: double): void

Words

@Test	new	pear	expected	actual	double	testPriceGetSet
Pear	Pear()	getPrice	setPrice	assertEquals	delta	0.001
String	message	"Pear price get set not working"				

```

@Test
public void testPriceGetSet() {
    Pear pear = new Pear();
    double expected = 4.0;
    double delta = 0.001;
    String message = "Pear price get set not working";
    pear.setPrice(expected);
    double actual = pear.getPrice();
    assertEquals(message, expected, actual, delta);
    //see multiple choice #23.
}

```

6. Using the words provided as a guide and the starter code re-write the main method presented so that if the user enters the phrase "tuna fish" the program will report the exception's message and then exit rather than crashing. Note: Need to use exception handling.
(Most of the words below only need to be used once)

[5 points]

```
import java.util.Scanner; // no need to rewrite the import
```

```
public class WeightProgram{
    public static void main(String[] args){
        Scanner s = new Scanner(System.in);
        System.out.println("What is the weight of the sample?");
        double d = s.nextDouble();
        System.out.println("Weight is: " + d);
    }
}
```

Words

try	Exception	getMessage	catch	e
----------------	-----------	------------	------------------	--------------

```
public static void main (String [] args) {
    try {
        Scanner s = new Scanner (System.in);
        System.out.println ("What is the weight of the sample?");
        double d = s.nextDouble();
        System.out.println ("Weight is: " + d);
    } //end try.
    catch (Exception e) {
        System.out.println (e.getMessage());
    }
} //end try-catch
} //end main
```