QUEEN MARY AND WESTFIELD COLLEGE
DCS/115 INTRODUCTION TO PROGRAMMING

Class Test Wednesday 7th February 2001 09.05-09.55

Please fill in your name here: **MODEL ANSWERS**

College registration number here:

Email address here: Course Code here:

All answers to this test should be written on the test sheet, but you may use spare paper for rough working, or the space on the back page if you cannot fit your answers into the space provided. Answer as many of the questions as you can. This is an "open book" test: you are allowed to access your notes and textbook.

1) In the two cases below suppose we start each time with a `String` variable `str` which refers to the string `"Hello world, how are you?"`. Write what would be printed when the code is executed:

a)

```
StringTokenizer tok =
   new StringTokenizer(str);
String acc="";
while(tok.hasMoreTokens()) {
    String t=tok.nextToken();
    if(t.charAt(1)=='o')
       acc+=t;
    acc+="*";
   }
System.out.println(acc);
```

*world,*how**you?*

b)

```
String acc="";
int i,j;
for(i=0; str.charAt(i)!='d';i++);
i+=2;
for(j=i; str.charAt(j)!='y'; j++)
    acc=str.charAt(j)+acc;
System.out.println(acc);
```

era woh

2)

Suppose we have an object referred to by variable `producer` which has a method with signature

```
public String make()
throws PException
```

Also a class called `Thing` which has a constructor with signature

```
public Thing(String str)
```

and `Thing` has a method with signature:

```
public void doIt()
throws TException
```

To the right, write a code fragment which produces a `Thing` object, using as the constructor argument the string obtained by calling `make()` on `producer`, then keeps calling `doIt()` on it until a `TException` occurs. This whole action should then be repeated until a `PException` occurs when `make()` is called.

```
while(true)
   try
      {
       Thing t = new
         Thing(producer.make());
       while(true)
          try
             {
              t.doIt();
             }
          catch(TException e)
             {
              break;
             }
      }
   catch(PException e)
      {
       break;
      }
```

3) The class `DogBot` to the left below models a simple artificial dog. Suppose we have two variables `rover` and `patch` of type `DogBot`. For each code fragment on the right, write what it will print:

```
class DogBot
{
 int hungry,tired;

 public DogBot(int h, int t)
 {
  hungry=h;
  tired=t;
 }

 public boolean eat()
 {
  if(hungry>6)
   {
    hungry-=3;
    return true;
   }
  else
     return false;
 }

 public void rest()
 {
  hungry++;
  tired-=2;
 }

 public void play()
 {
  hungry+=2;
  tired+=3;
 }

 public String noise()
 {
  if(hungry>8&&tired<11)
     return "whine ";
  else
      if(tired>7&&tired>hungry)
        return "snore ";
     else
        return "woof ";
 }
}
```

a)
```
rover = new DogBot(6,4);
rover.play();
System.out.print(rover.noise());
rover.play();
System.out.print(rover.noise());
rover.rest();
System.out.println(rover.noise());
```

woof whine whine

b)
```
rover = new DogBot(7,8);
if(rover.eat())
   rover.rest();
if(rover.eat())
   System.out.print(rover.noise());
rover.play();
System.out.println(rover.noise());
```

snore

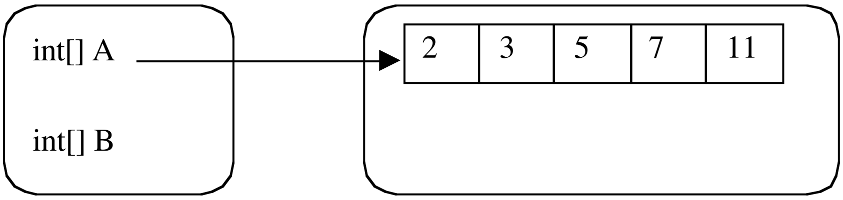
c)
```
rover = new DogBot(2,1);
patch = new DogBot(7,5);
rover.play();
if(patch.eat())
   patch.play();
System.out.print(rover.noise());
System.out.println(patch.noise());
```

woof snore

d)
```
rover = new DogBot(5,4);
rover.play();
patch = rover;
if(patch.eat())
   System.out.print(patch.noise());
rover.play();
System.out.print(rover.noise());
System.out.println(patch.noise());
```

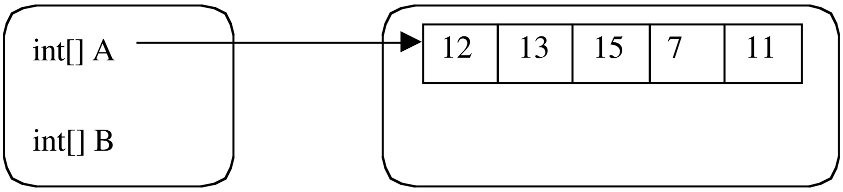
woof snore snore

4) Assume the following picture shows the state of the store before the execution of each sequence of instructions below (i.e. the situation is returned to this state before the execution of each part).

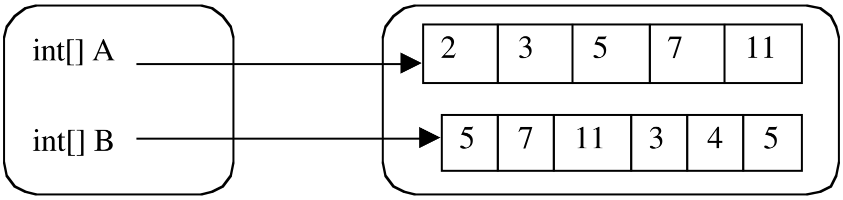


Then, for each of parts (a) to (d) below complete the picture of the state of the store after the execution of the given program.

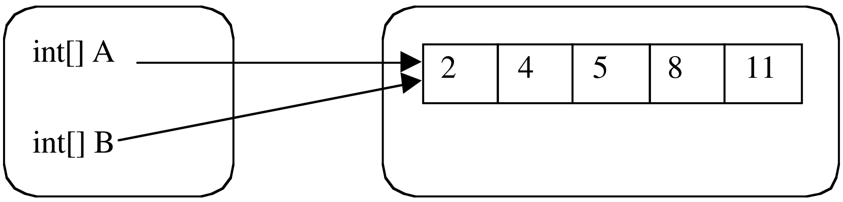
a)
```
for (int i=0; A[i]<6; i++)
    A[i] += 10;
```



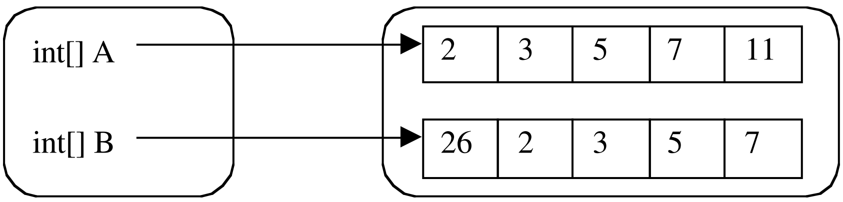
b)
```
B = new int[6];
for (int i=0; i<3; i++)
    B[i] = A[i+2];
for (int j=3; j<6; j++)
    B[j] = j;
```



c)
```
B = A;
for (int i=1; i<5; i++)
    if (B[i]!=5)
        B[i]++;
A[4]--;
```



d)
```
B = new int[5];
int sum=0;
for (int i=1; i<5; i++) {
     sum+=A[i];
     B[i] = A[i-1];
}
B[0]=sum;
```



Please use the space here if you need any additional space to answer any of the questions. For anything written here, please indicate which question it applies to.