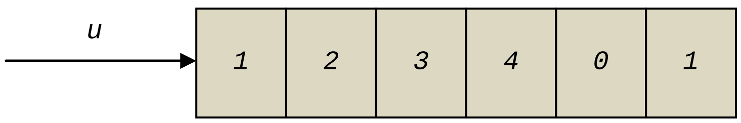
Exam – required part: solution

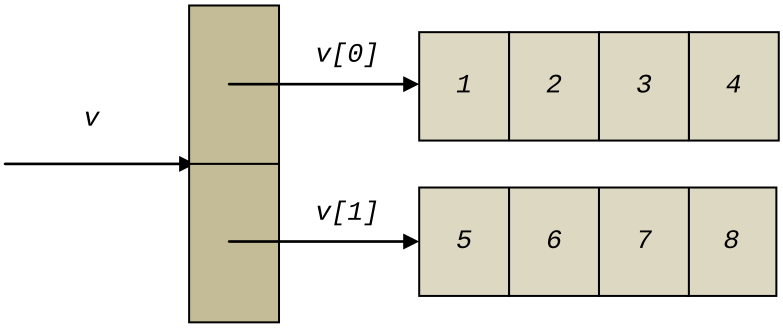
Tasks: solutions

Task 1 (2 points + 2 points)

a) (2 points)



b) (2 points)



Task 2 (2 points + 2 points + 2 points)

a) (2 points)

When you execute the code fragment an exception is thrown:

`java.lang.ArithmeticException: / by zero`

The denominator `u.length - 4` is `0`, but division by `0` is not allowed.

b) (2 points)

When you execute the code fragment an exception is thrown:

`java.lang.ArrayIndexOutOfBoundsException: 4`

There is an attempt to use `u.length`, i.e. `4`, as the index. But the greatest index in the array `u` is `3` – index `4` does not exist.

c) (2 points)

When you execute the code fragment an exception is thrown:

`java.lang.NullPointerException`

The reference `v[0]` is `null`, it does not refer to any object. This means that the reference cannot activate the method `append`.

Task 3 (4 points + 3 points)

a) (4 points)

```
public static Integer[] filter (Integer[] v, Integer p)
{
    // the number of integers smaller than the given integer
    int    countIntegers = 0;
    for (Integer i : v)
        if (i.lessThan (p))
            countIntegers++;

    // the integers that are smaller than the given integer
    Integer[]    w = new Integer[countIntegers];
    int    pos = 0;
    for (Integer i : v)
        if (i.lessThan (p))
            w[pos++] = i;

     return w;
}
```

b) (3 points)

```
Integer[]    v = { new Integer (40),
                   new Integer (100),
                   new Integer (50),
                   new Integer (10),
                   new Integer (70)
                 };
Integer    p = new Integer (70);
Integer[]    w = filter (v, p);

for (Integer i : w)
    System.out.print (i);
System.out.println ();
```

Task 4 (2 points + 2 points + 2 points + 2 points)

a) (2 points)

```
public String toString ()
{
    StringBuilder    sb = new StringBuilder ("{");
    sb.append (startPoint);
    sb.append (", ");
    sb.append (endPoint);
    sb.append ("}");

    return sb.toString ();
}
```

b) (2 points)

```
public double length ()
{
    return startPoint.distance (endPoint);
}
```

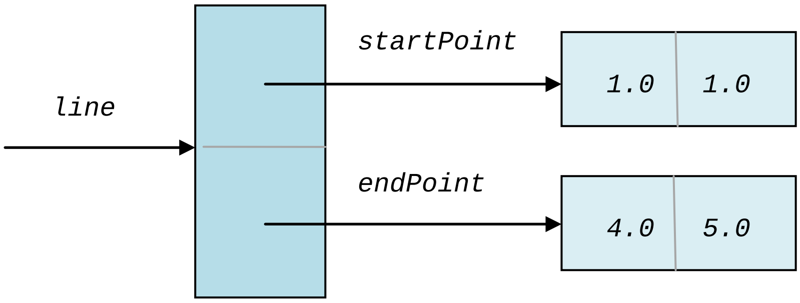
c) (2 points)

```
public Point midPoint ()
{
    double    midX = (startPoint.x + endPoint.x) / 2;
    double    midY = (startPoint.y + endPoint.y) / 2;
    Point    midP = new Point (midX, midY);

    return midP;

}
```

d) (2 points)



Task 5 (4 points + 2 points + 2 points)

a) (4 points)

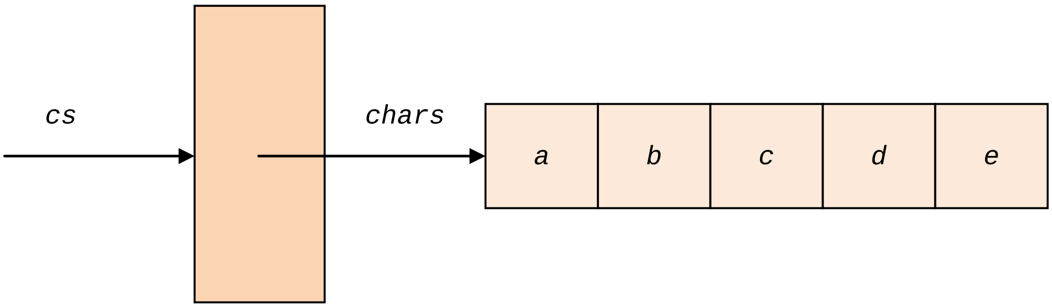
```
// charAt returns the character located at a given index in the sequence
public char charAt (int index)
{
    return this.chars[index];
}

// length returns the length of the character sequence
public int length ()
{
    return this.chars.length;
}
```

b) (2 points)

If statement (1) is included there will be a compilation error. The reference `cs` is of type `CharSequence` and it can only activate the methods in the interface `CharSequence` (the methods `charAt` and `length`). It cannot activate the method `startsWith`. That method can be activated with a reference of type `ImmutableCharSequence`.

c) (2 points)


cs
chars
a
b
c
d
e