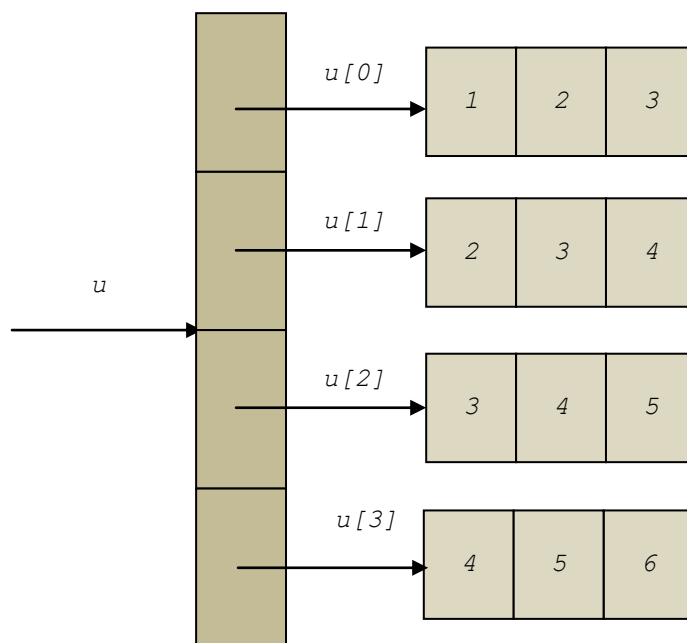


Exam – required part: solution

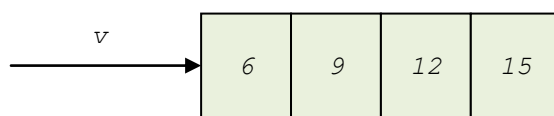
Tasks: solutions

Task 1 (1 point + 3 points)

a) (1 point)



b) (3 points)



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

a) (2 points)

```
if (n2 < n3)
{
    if (n2 < n4)
        m = n2;
    else
        m = n4;
}
else
```

```

{
    if (n3 < n4)
        m = n3;
    else
        m = n4;
}

```

b) (2 points)

```

public static int minShort (int n1, int n2, int n3, int n4)
{
    int    m = n1;
    if (n2 < m) m = n2;
    if (n3 < m) m = n3;
    if (n4 < m) m = n4;

    return m;
}

```

c) (2 points)

In an execution of method `minLong` there are 3 comparisons performed.

In an execution of method `minShort` there are also 3 comparisons performed.

Task 3 (4 points + 3 points)

a) (4 points)

```

public static Country[] filter (Country[] countries, char firstLetter)
{
    // determine the number of acceptable countries
    int    numberOfAcceptableCountries = 0;
    for (Country c : countries)
        if (c.name ().charAt (0) == firstLetter)
            numberOfAcceptableCountries++;

    // determine the acceptable countries
    Country[]    acceptedCountries = new Country[numberOfAcceptableCountries];
    int    posAcceptedCountries = 0;
    for (Country c : countries)
        if (c.name ().charAt (0) == firstLetter)
            acceptedCountries[posAcceptedCountries++] = c;

    return acceptedCountries;
}

```

b) (3 points)

```

Country[]    countries = { new Country ("Austria", "Wien"),
                           new Country ("Spain", "Madrid"),
                           new Country ("Italy", "Rome"),
                           new Country ("Bulgaria", "Sofia"),
                           new Country ("Sweden", "Stockholm")
                           };

Country[]    acceptedCountries = filter (countries, 'S');

```

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

a) (2 points)

```

public String toString ()

```

```

{
    StringBuilder sb = new StringBuilder ("{ ");
    sb.append (vertex1.toString ().substring (14) + " ");
    sb.append (vertex2.toString ().substring (14) + " ");
    sb.append (vertex3.toString ().substring (14) + " ");
    sb.append ("}");

    return sb.toString ();
}

```

b) (2 points)

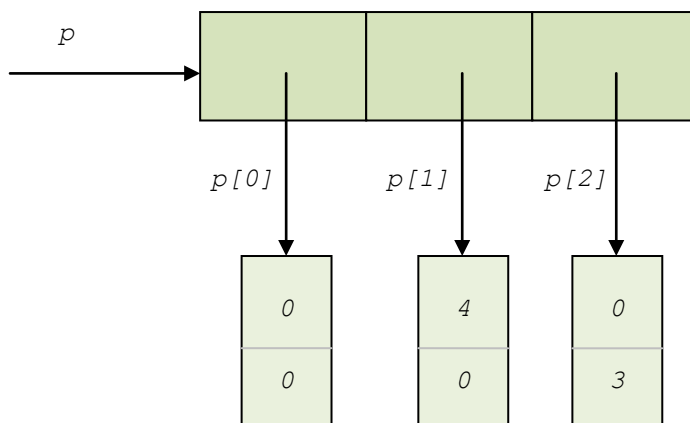
```

public double perimeter ()
{
    double per = vertex1.distance (vertex2) +
                vertex2.distance (vertex3) +
                vertex3.distance (vertex1);

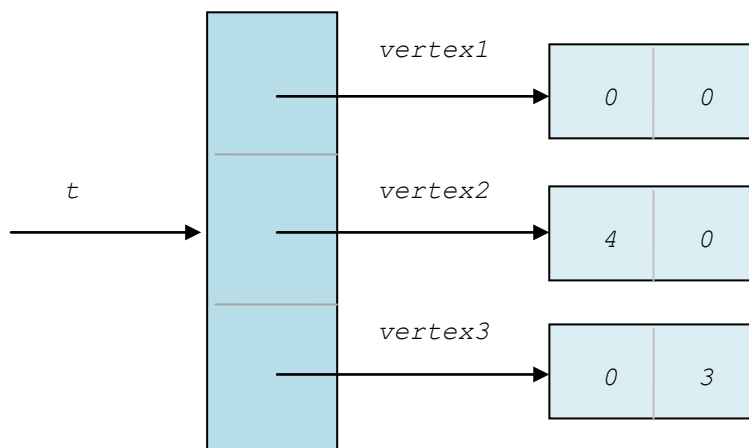
    return per;
}

```

c) (2 points)



d) (2 points)



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

a) (2 points)

```
public ColouredCircle (Point centre, double radius, String colour)
{
    super (centre, radius);
    this.colour = colour;
}
```

b) (2 points)

```
public String toString ()
{
    StringBuilder sb = new StringBuilder ("{");
    sb.append (super.toString () + ", ");
    sb.append (colour);
    sb.append ("}");

    return sb.toString ();
}
```

c) (2 points)

If statement (1) is included there is a compilation error. The method `getColour` is not present in the superclass `Circle`, and cannot be activated by a reference to type `Circle`.

If statement (2) is included, the area of the coloured circle is also printed. The method `area` is inherited from the superclass `Circle`, and a reference of type `ColouredCircle` can therefore be used to activate that method.

d) (2 points)

