

Question 1 (5 points)

For the following questions, circle the correct answer. Every correct answer gives +1 point, every wrong answer gives -0.5 points, no answer gives 0 points.

- (i) How many asterisks (symbol “*”) are printed by the following piece of code?

```
for (int i=3;i<9;i++) {  
    System.out.print("*");  
}
```

- (a) 6
 - (b) 7
 - (c) 9
 - (d) 10
 - (e) compiler error
- (ii) Which of the following is an infinite loop?
- (a) `for (;;) //some code here`
 - (b) `for (int i=10;i>0;i++) //some code here`
 - (c) `for (int i=0;;i++) //some code here`
 - (d) all of the above
 - (e) none of the above
- (iii) How many basic data types exist in Java?
- (a) 2
 - (b) 4
 - (c) 8
 - (d) 16
 - (e) 32
- (iv) What is *byte code* in Java?
- (a) block of code written in Java
 - (b) code generated by a Java Virtual Machine (JVM)
 - (c) code generated by a Java Compiler
 - (d) name of a Java source file
 - (e) another name for the compiler
- (v) What will be the result of the method call `square(5)` if the code for the `square` method is the one given below:

```
public static int square(int n) {  
    if (n==1)  
        return 1;  
    else  
        return square(n-1) + 2*n - 1;  
}
```

- (a) 1
- (b) 4
- (c) 16
- (d) 25
- (e) infinite run

Question 2 (8 points)

What is type widening and narrowing? How do you do this in Java? Give two examples (one for widening and one for narrowing) of when this is useful/necessary.

Question 3 (8 points)

Explain briefly what is happening at each step of the “fetch, decode, execute” cycle (or instruction cycle) and which part(s) of the computer is(are) involved in each step.

Question 4 (7 points)

You are given the following program in Java. What will be printed in the output?

```
import java.util.Arrays;

public class FancyClass {

    public static void fancy(int a, int b, int[] list) {
        int temp;
        temp = list[a];
        list[a] = list[b];
        list[b] = temp;
    }

    public static void main(String[] args) {
        int value1 = 2;
        int value2 = 3;
        int[] list = {1,3,5,7,9};

        fancy(value1, value2, list);

        System.out.println(value1);
        System.out.println(value2);
        System.out.println(Arrays.toString(list));

        fancy(list[0], list[1], list);

        System.out.println(value1);
        System.out.println(value2);
        System.out.println(Arrays.toString(list));

        fancy(value1, list[value1], list);

        System.out.println(value1);
        System.out.println(value2);
        System.out.println(Arrays.toString(list));

    }
}
```

Question 5 (7 points)

A student (newbie in programming) is trying to assess whether the following code will work or not. Can you help? If you think the code works as it is, explain shortly why, otherwise point out the syntax error(s) by naming the specific problem(s) and the relevant line(s).

```
1: public class surprise {  
2:     public static void main(String[] args) {  
3:         play(1,2,3);  
4:         play(1,2);  
5:         int x = play(5,6,7);  
6:     }  
7:  
8:     public static void play(int a, int b, int c) {  
9:         System.out.println(a+b+c);  
10:    }  
11:  
12:    public static void play(int a, int b) {  
13:        System.out.println(a+b);  
14:    }  
15:  
16:    public static int play(int a, int b, int c) {  
17:        return (a+b+c);  
18:    }  
19: }
```

Question 6 (15 points)

Write a method `multiplySquareMatrices` to calculate the dot product of two **square** matrices with the same size that takes as parameters both matrices (integers) and returns the resulting one. The code can assume that this requirement is met (i.e. you do not need to do any check to the dimensions of the matrices). The dot product of two square matrices can be summarized as a summation of (row-column) products as follows.

Considering the matrices A and B as:

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}; \quad B = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$$

$$\text{The dot product } A \cdot B = \begin{bmatrix} a_{11} * b_{11} + a_{12} * b_{21} & a_{11} * b_{12} + a_{12} * b_{22} & a_{11} * b_{13} + a_{12} * b_{23} \\ a_{21} * b_{11} + a_{22} * b_{21} & a_{21} * b_{12} + a_{22} * b_{22} & a_{21} * b_{13} + a_{22} * b_{23} \end{bmatrix}$$

Similarly, for two 3x3 matrices:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}; \quad B = \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix}$$

The dot product $A \cdot B =$

$$\begin{bmatrix} a_{11} * b_{11} + a_{12} * b_{21} + a_{13} * b_{31} & a_{11} * b_{12} + a_{12} * b_{22} + a_{13} * b_{32} & a_{11} * b_{13} + a_{12} * b_{23} + a_{13} * b_{33} \\ a_{21} * b_{11} + a_{22} * b_{21} + a_{23} * b_{31} & a_{21} * b_{12} + a_{22} * b_{22} + a_{23} * b_{32} & a_{21} * b_{13} + a_{22} * b_{23} + a_{23} * b_{33} \\ a_{31} * b_{11} + a_{32} * b_{21} + a_{33} * b_{31} & a_{31} * b_{12} + a_{32} * b_{22} + a_{33} * b_{32} & a_{31} * b_{13} + a_{32} * b_{23} + a_{33} * b_{33} \end{bmatrix}$$

Example 1:

$$A = \begin{bmatrix} 1 & 2 \\ 2 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 3 & 1 \\ 1 & 4 \end{bmatrix}$$

$$A \cdot B = \begin{bmatrix} (1 * 3 + 2 * 1) & (1 * 1 + 2 * 4) \\ (2 * 3 + 0 * 1) & (2 * 1 + 0 * 4) \end{bmatrix} = \begin{bmatrix} 5 & 9 \\ 6 & 2 \end{bmatrix}$$

Example 2:

$$A = \begin{bmatrix} 1 & 0 & 1 \\ 2 & 1 & 2 \\ 3 & 2 & 3 \end{bmatrix}; \quad B = \begin{bmatrix} 0 & 1 & 2 \\ 1 & 2 & 3 \\ 2 & 3 & 4 \end{bmatrix}$$

$$A \cdot B = \begin{bmatrix} 2 & 4 & 6 \\ 5 & 10 & 15 \\ 8 & 16 & 24 \end{bmatrix}$$

Provide your code in this box:

Question 7 (15 points)

Write a method `hasSameElements` that takes as parameters two 1-dimensional arrays of integers and checks if all the elements of the arrays are equal **regardless of their order**. The method returns `true` if all the elements of one matrix appear in the other one and vice-versa or `false` otherwise. Notice that elements value can be repeated more than once.

- *Example 1:*

`arrayA = 0,1,2,3,0;`

`arrayB = 1,3,2,0,5;`

The method must return `false` since number 5 is not an element of arrayA.

- *Example 2:*

`arrayA = 0,1,5,2,0;`

`arrayB = 1,0,2,0,5;`

The method must return `true` since all the elements in vectorA are also elements of vectorB (notice that there are two elements with value 0 **in both vectors**).

Question 8 (15 points)

Dutch vaccination plan has been heavily criticised, so the government once again needs your help with a new vaccine. One doctor has isolated the RNA of COVID-19, and as you hopefully know (and remember) from basic biology that is a sequence of nitrogenous bases (adenine (A), guanine (G), thymine (T), and cytosine (C)).

For simplicity reasons we store the sequence of RNA in an integer array where 1 corresponds to A, 2 corresponds to G, 3 corresponds to T and 4 corresponds to C. In order to construct the vaccine, doctors need to re-construct the RNA sequence such that all the same nitrogenous bases appear one after the other in the sequence, i.e. all 1 must be together, all 2 must be together etc. in that particular order, i.e. all 1s come first and all 4s come last.

Write a method called `vaccineRNA` that takes as parameter a 1-dimensional array called `RNA` and returns a 1-dimensional array that contains the transformed RNA sequence such that all the same nitrogenous bases are together. The original array must remain unchanged.

- *Example 1:*
RNA = 1,1,2,3,2,4,4,4,1,2,4;
The method must return the array 1,1,1,2,2,2,3,4,4,4,4.
- *Example 2:*
RNA = 4,4,4,4,3,3,3,3,2,2,2,2,1,1,1,1;
The method must return the array 1,1,1,1,2,2,2,2,3,3,3,3,4,4,4,4.

Student name:
Student ID:

KEN1120 (2020/2021)

Student name:
Student ID:

KEN1120 (2020/2021)
