ADVANCE PROGRAMMING (CSE201)

MID SEM EXAM

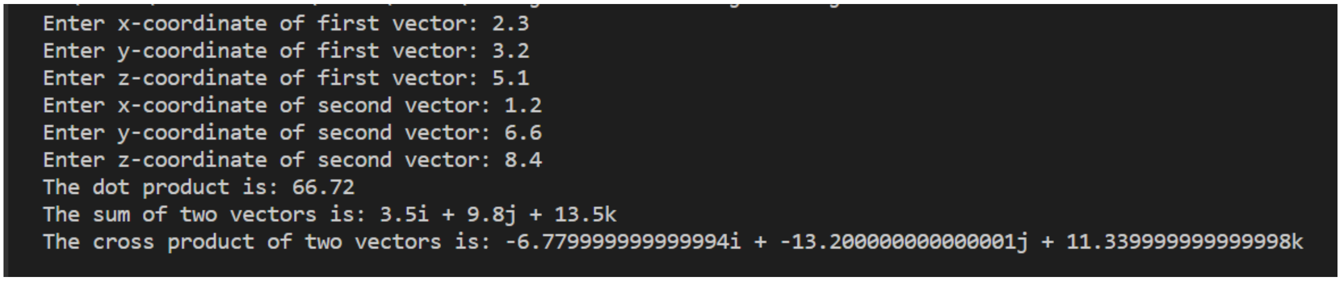
Monsoon 2022

QUESTION-1: Create a class Vector that has 3 double attributes x_coord, y_coord and z_coord. Add constructors/getters/setters as you like. In this class, also write methods to add 2 vectors, calculate the dot product of 2 vectors, and calculate the cross product of 2 vectors. These methods should take 2 Vector objects as arguments (also note that these methods are NOT called with respect to a particular object). In the Main class, take 6 numbers as user input, and use them to create 2 objects of class Vector. Perform addition, dot product and cross product on these 2 vectors, and print the results.

NOTE:

1) Given vectors aî + bĵ + ck and xî + yĵ + zk, the dot product is a*x+b*y+c*z.
2) Given vectors a1î + a2ĵ + a3k and b1î + b2ĵ + b3k, the cross product is (b2a3-a2b3)î - (a1b3-b1a3)ĵ + (a1b2-a2b1)k. **(40% Weightage)**

TEST CASE:



```
Enter x-coordinate of first vector: 2.3
Enter y-coordinate of first vector: 3.2
Enter z-coordinate of first vector: 5.1
Enter x-coordinate of second vector: 1.2
Enter y-coordinate of second vector: 6.6
Enter z-coordinate of second vector: 8.4
The dot product is: 66.72
The sum of two vectors is: 3.5i + 9.8j + 13.5k
The cross product of two vectors is: -6.779999999999994i + -13.200000000000001j + 11.339999999999998k
```

QUESTION-2: You have to design an OOPS based system which makes use of concepts such as encapsulation, generics, inheritance and more (as applicable) with the following requirements -

1. **Fractions (represented as a/b where a and b are integers and b != 0)**
   - add two fractions
   - multiply two fractions
   - print a fraction in the format “a/b”
   - make sure each fraction is in reduced form, i.e. gcd(a,b) = 1 is ensured after every operation

2. **Complex numbers (represented as a + ib where i = sqrt(-1))**

   Note: ‘a’ and ‘b’ can either be integers or Fractions themselves and both cases should be handled separately. For this you have to create an abstract generic class **Complex** which will have two child classes **ComplexInteger** and **ComplexFraction** which will have different implementations for the following operations. For **ComplexFraction,** you have to make use of methods created in **Fraction** class

   - add two complex numbers

   - multiply two complex numbers
     Example: (1 + 2i) * (2 + i) = 1*2 + 1*i + 2i*2 + 2i*i = 2-2 + 5i = 5i

   - argument of a complex number
     Argument(a+ib) = tan_inverse(b/a) (you can use Math.atan function)

   - magnitude of a complex number
     Magnitude(a+ib) = sqrt(a*a + b*b)

   - print a complex number in the format “a + ib” or “a - ib” as appropriate. If ‘a’ and ‘b’ are fractions, use print from **Fraction** class to print ‘a’ and ‘b’

Note: you are free to use any other helper functions / methods / getters / setters aside the mandatory ones mentioned above.

Note: You have to write the code from scratch - you can either take values as input or hard code them. An example for your reference is given below. (**60% Weightage**)
—-

TEST CASE

(The objects are - ⅔, ¼, 2+3i, 1+2i, ⅓ + ⅘ i, ½ + ½ i)

6 // total number of numbers

fraction // number type is fraction

2 3 // numerator and denominator

fraction // number type is fraction

1 4 // numerator and denominator

complex // number type is complex number

integer // complex number has integral parts

2 3 // real and imaginary parts

complex // number type is complex

integer // it is an integer complex

1 2

complex

fraction // fraction based complex number

1 3 4 5 // numerator denominator of real part | numerator denominator of complex part

complex

fraction

1 2 1 2

<add two fractions>

11/12

<multiply two fractions>

3/2 // note that the fraction is in reduced form

<add integer complex numbers>

3 + 5i

<multiple integer complex numbers>

-4 + 7i

<argument of 2+3i>

56 degrees // answer in radians also acceptable

<magnitude of 2+3i>

3.606 // correct to 3 decimal places

<add fraction complex numbers>

5/6 + i 13/10

<multiply fraction complex numbers>

-7/30 + i 17/30

<argument of 1/2 + i 1/2>

45 degrees // answer in radians also acceptable

<magnitude of 1/2 + i 1/2>

0.707 // correct upto three decimal places