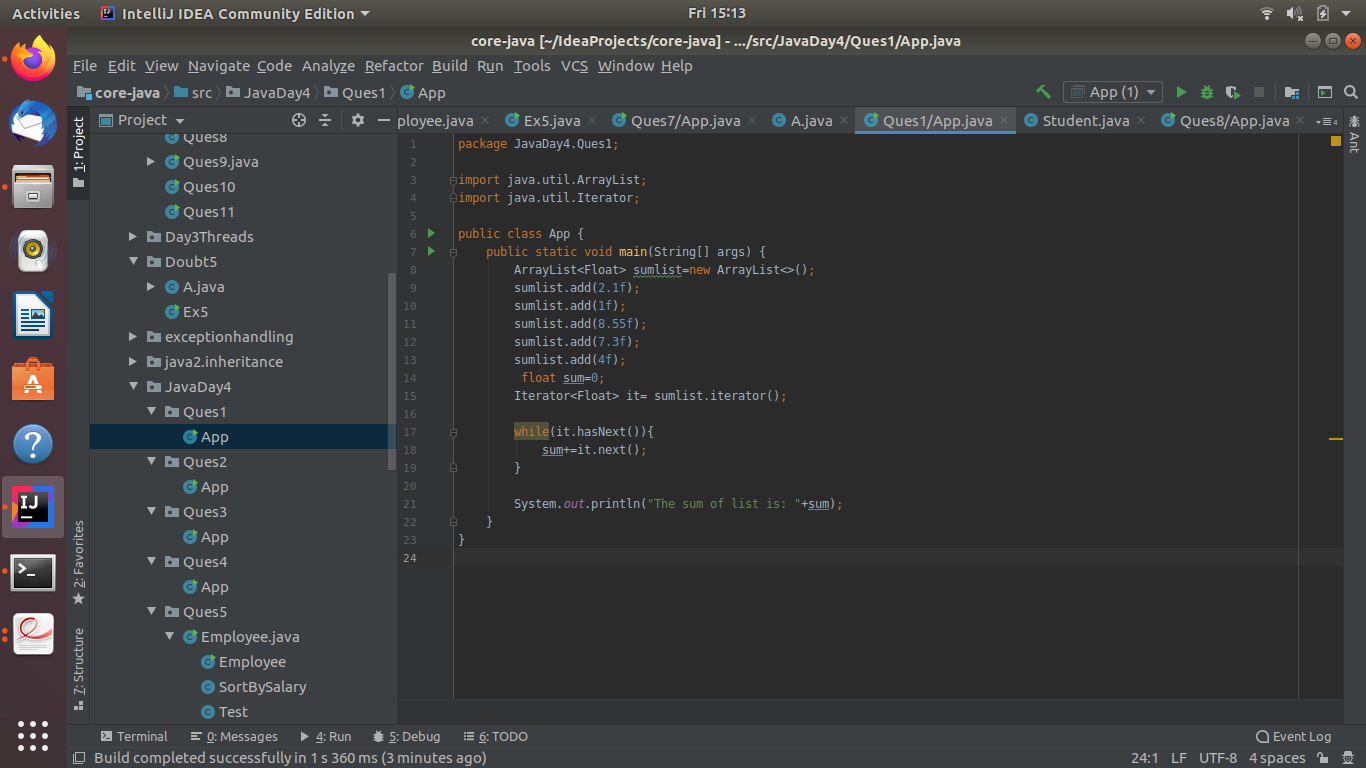
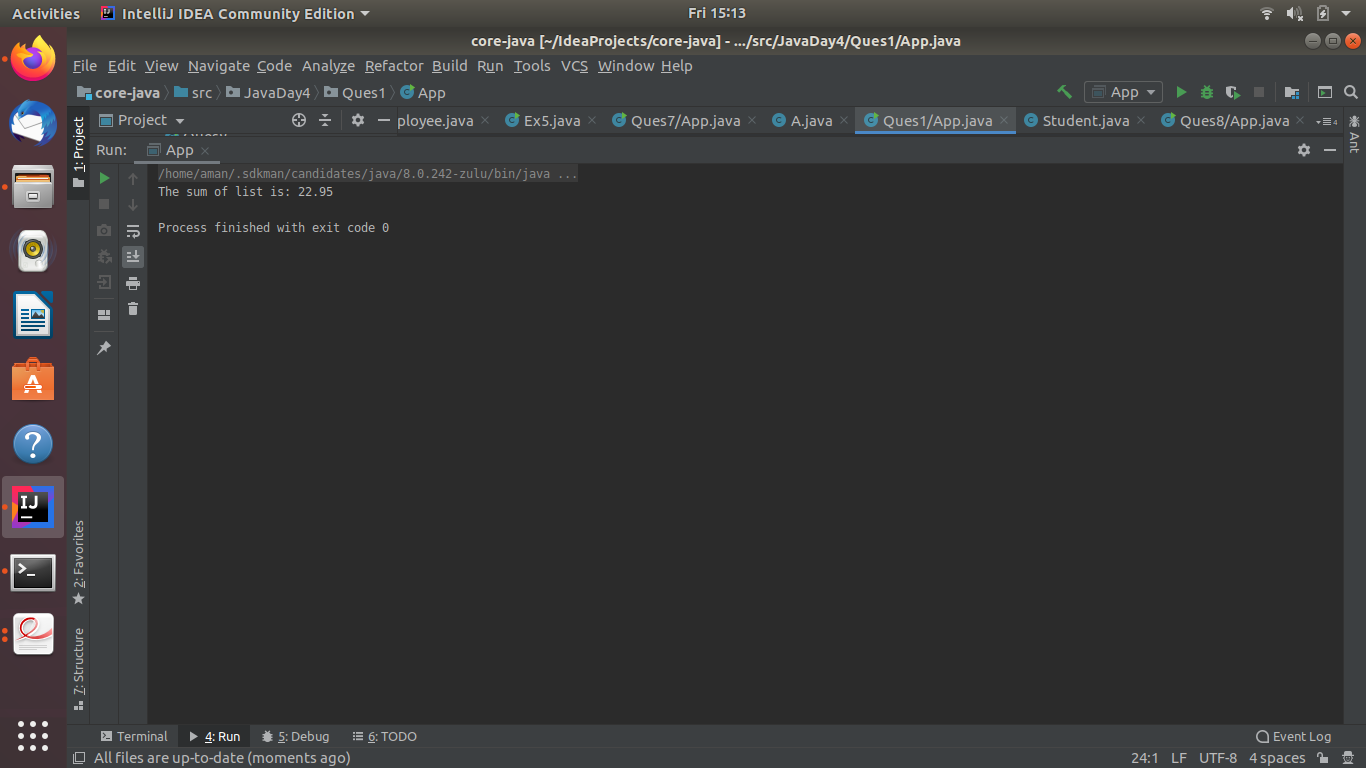
1. Write Java code to define List . Insert 5 floating point numbers in List, and using an iterator, find the sum of the numbers in List.

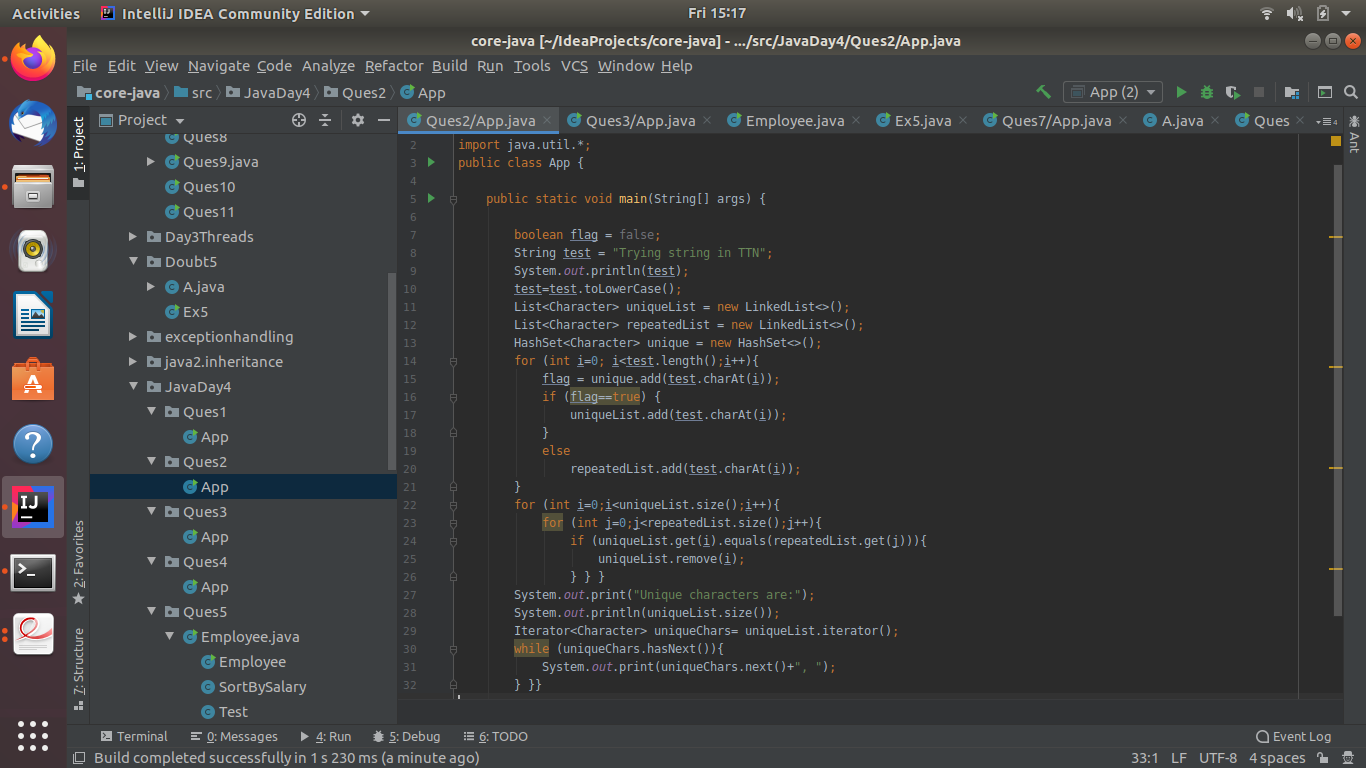
**Sol 1.**

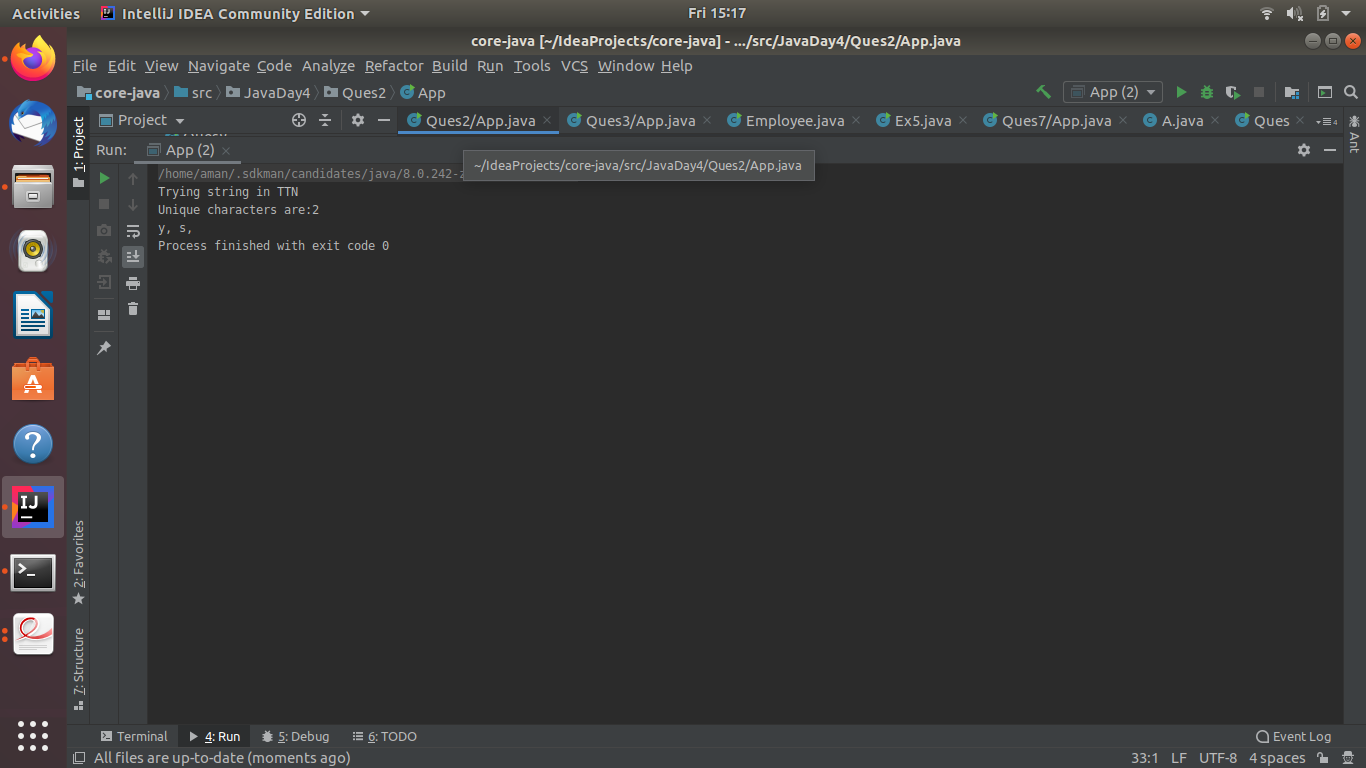
****

****

1. Write a method that takes a string and returns the number of unique characters in the string.

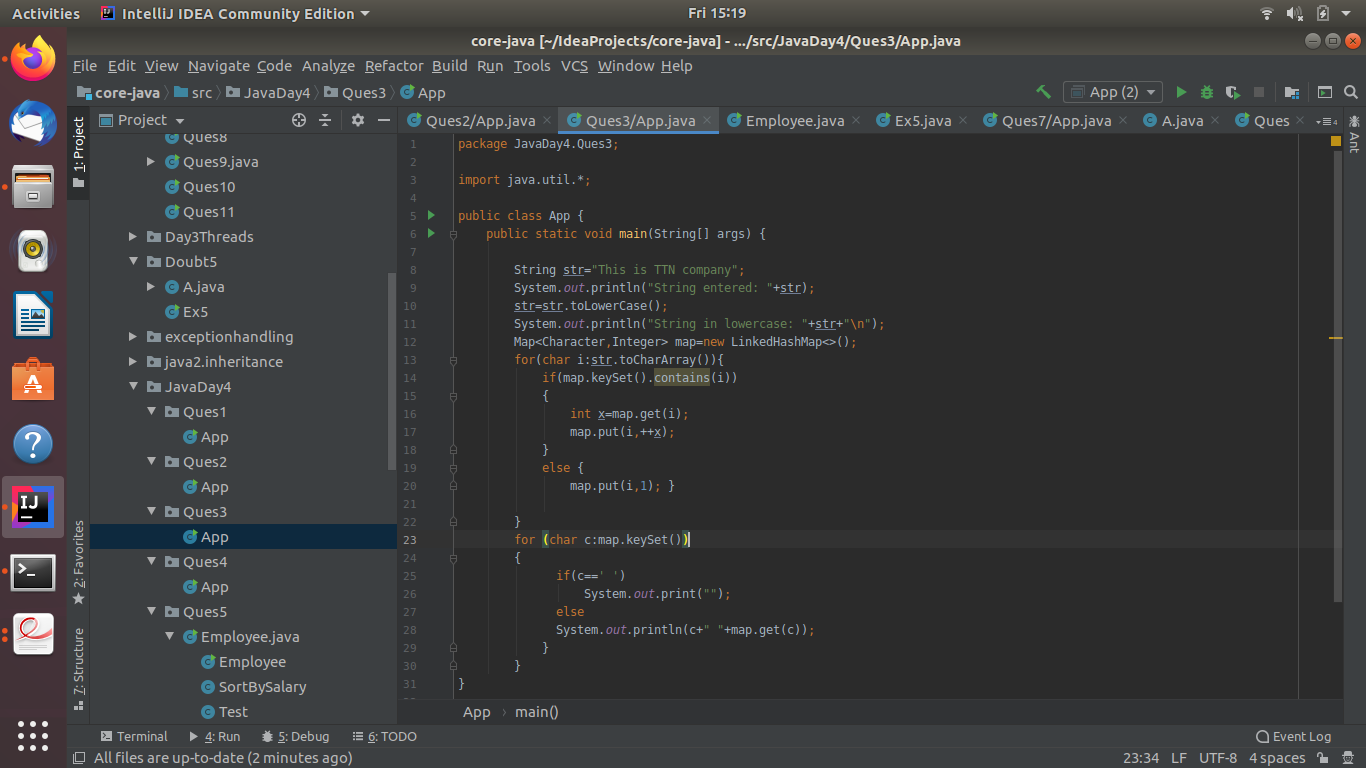
**Sol 2.**

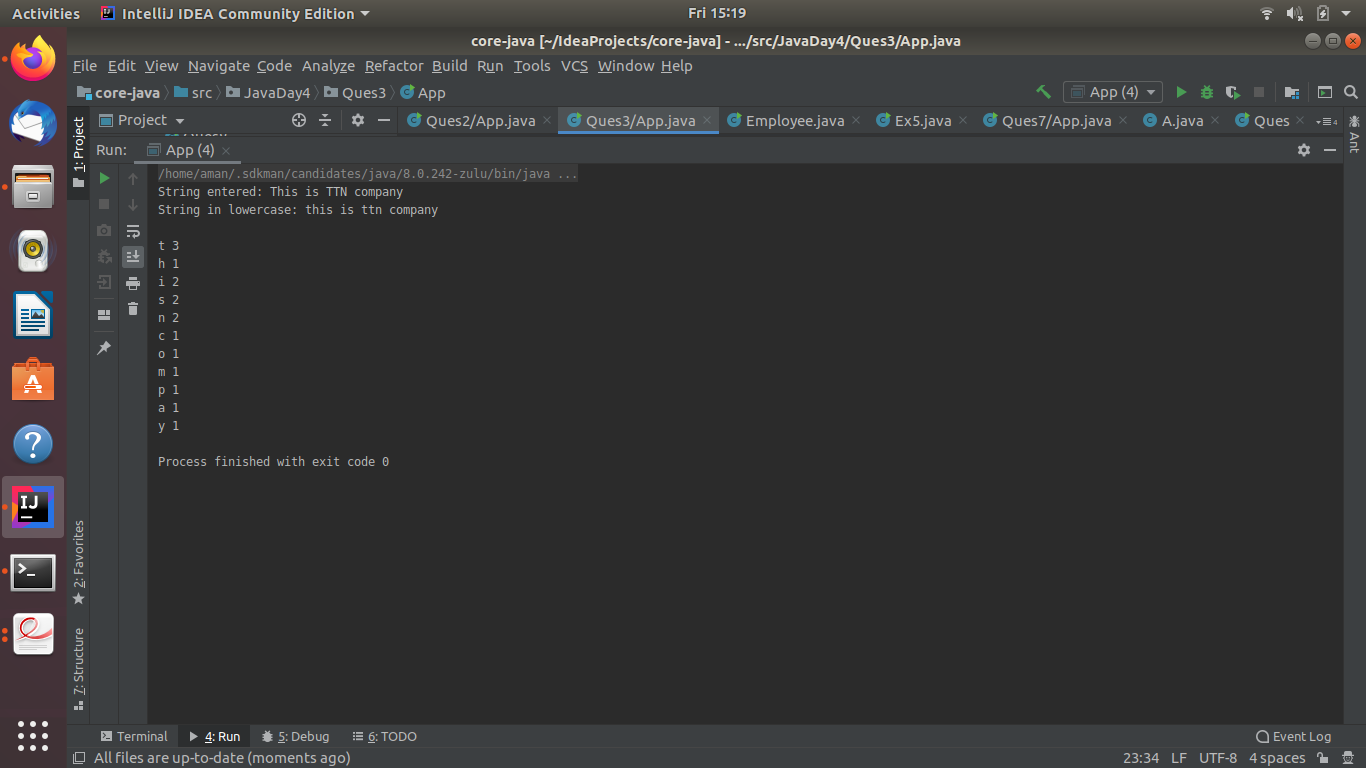
****

****

1. Write a method that takes a string and print the number of occurrence of each character characters in the string.

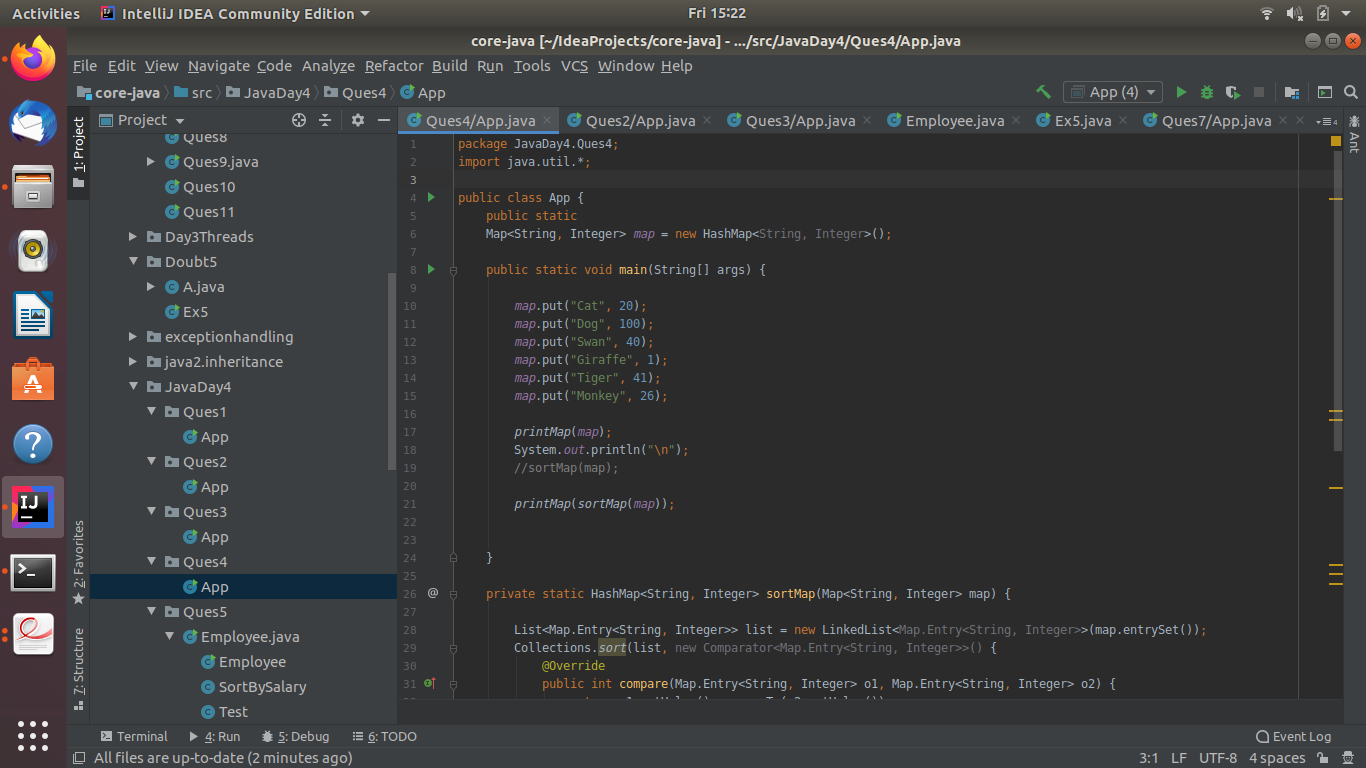
**Sol 3.**

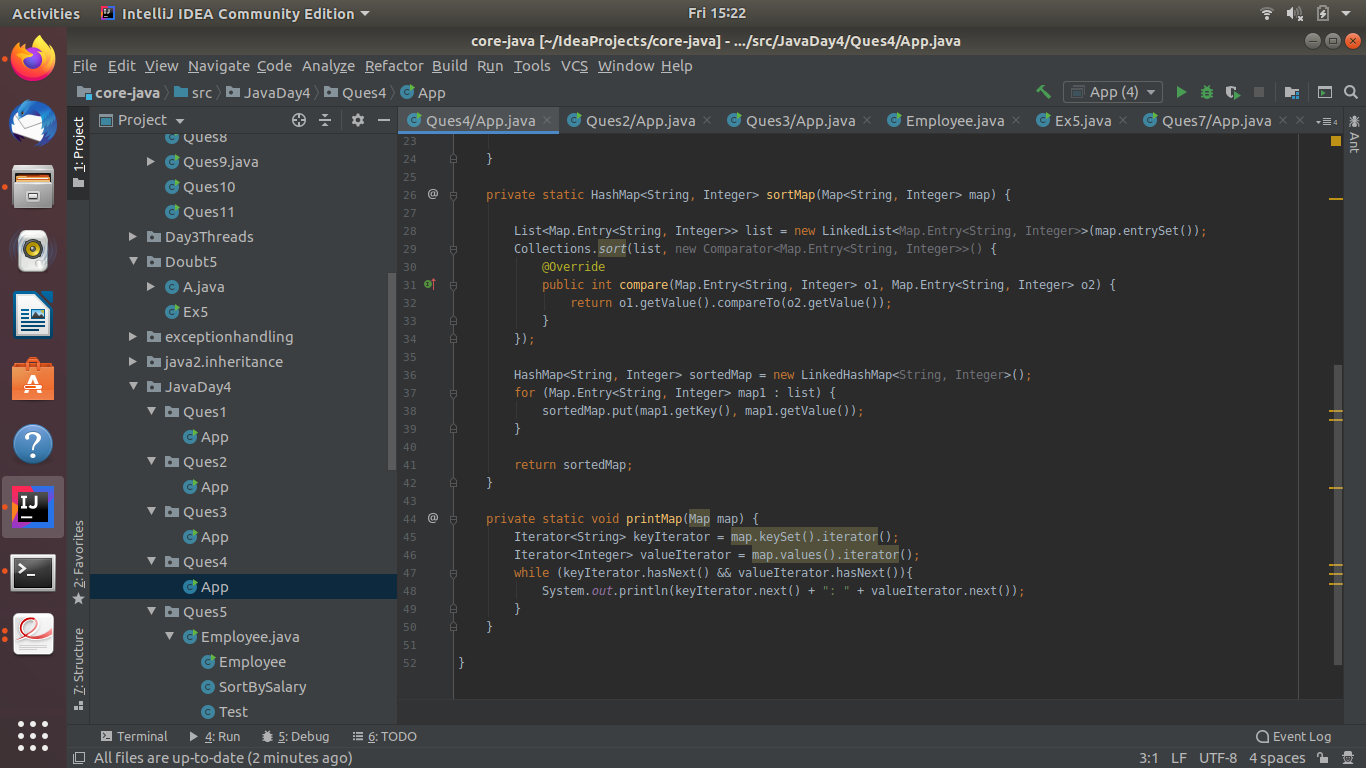
****

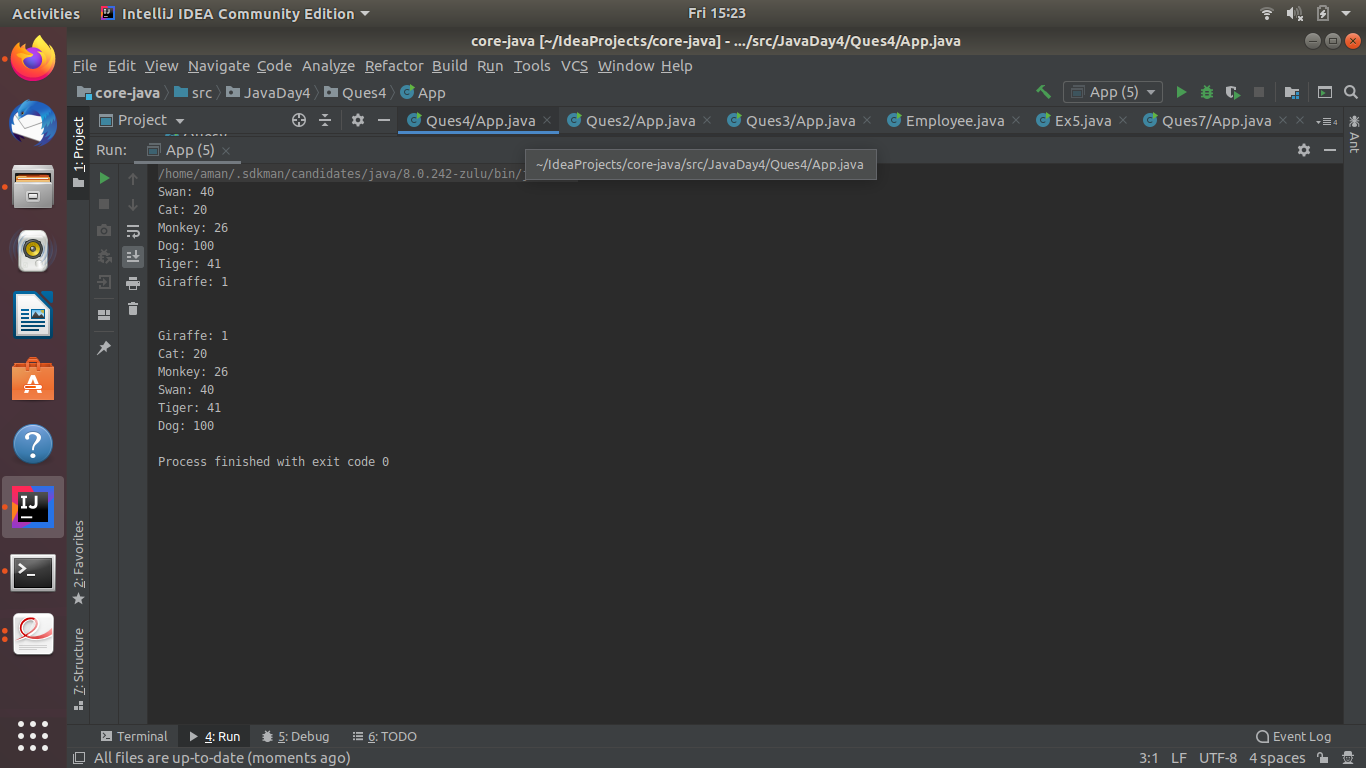
****

1. Write a program to sort HashMap by value.

**Sol 4.**

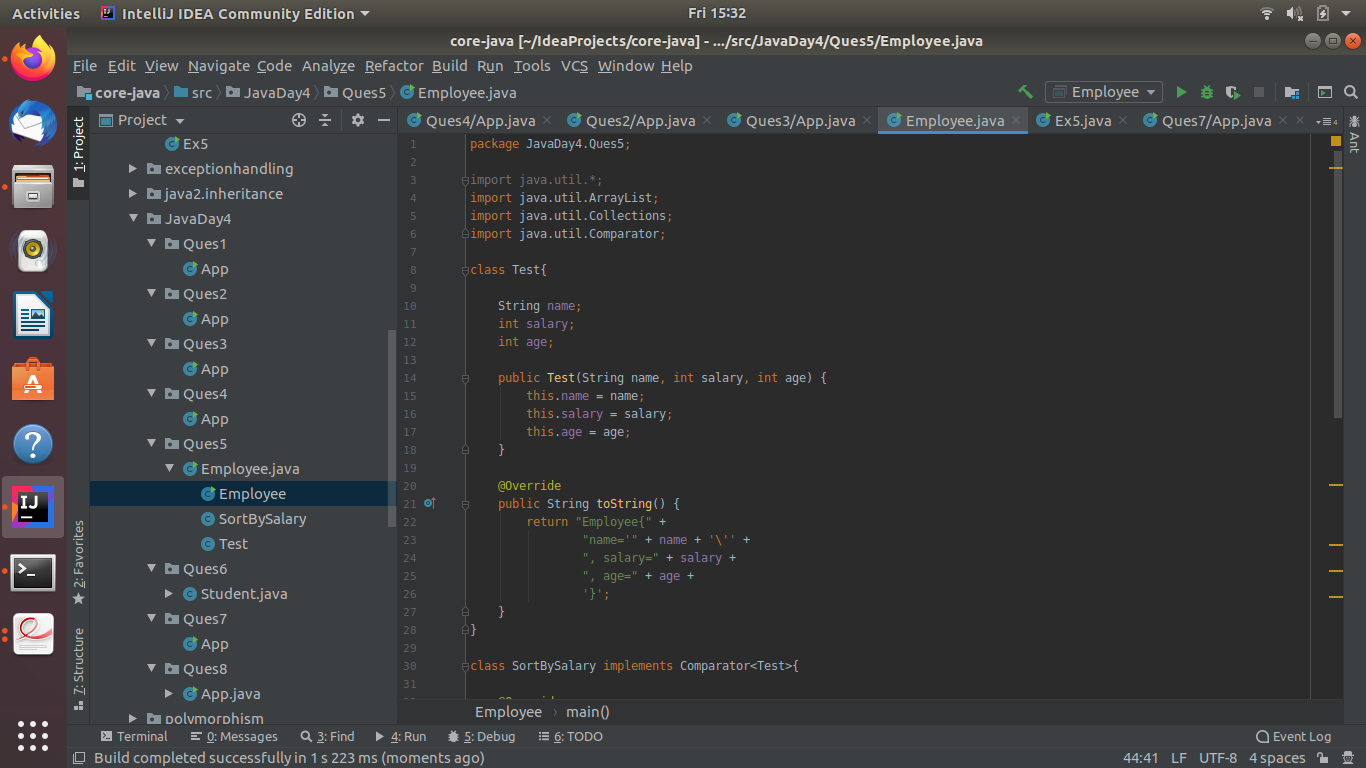
****

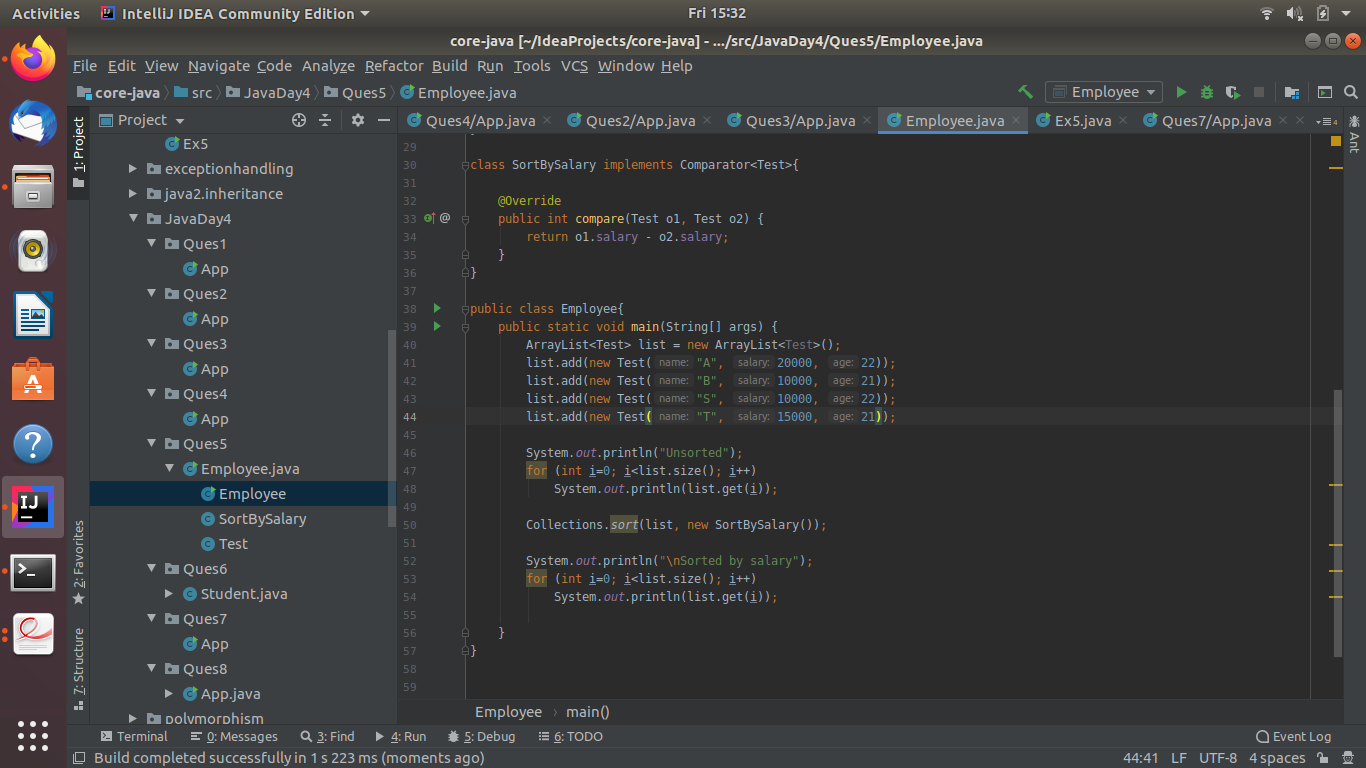
****

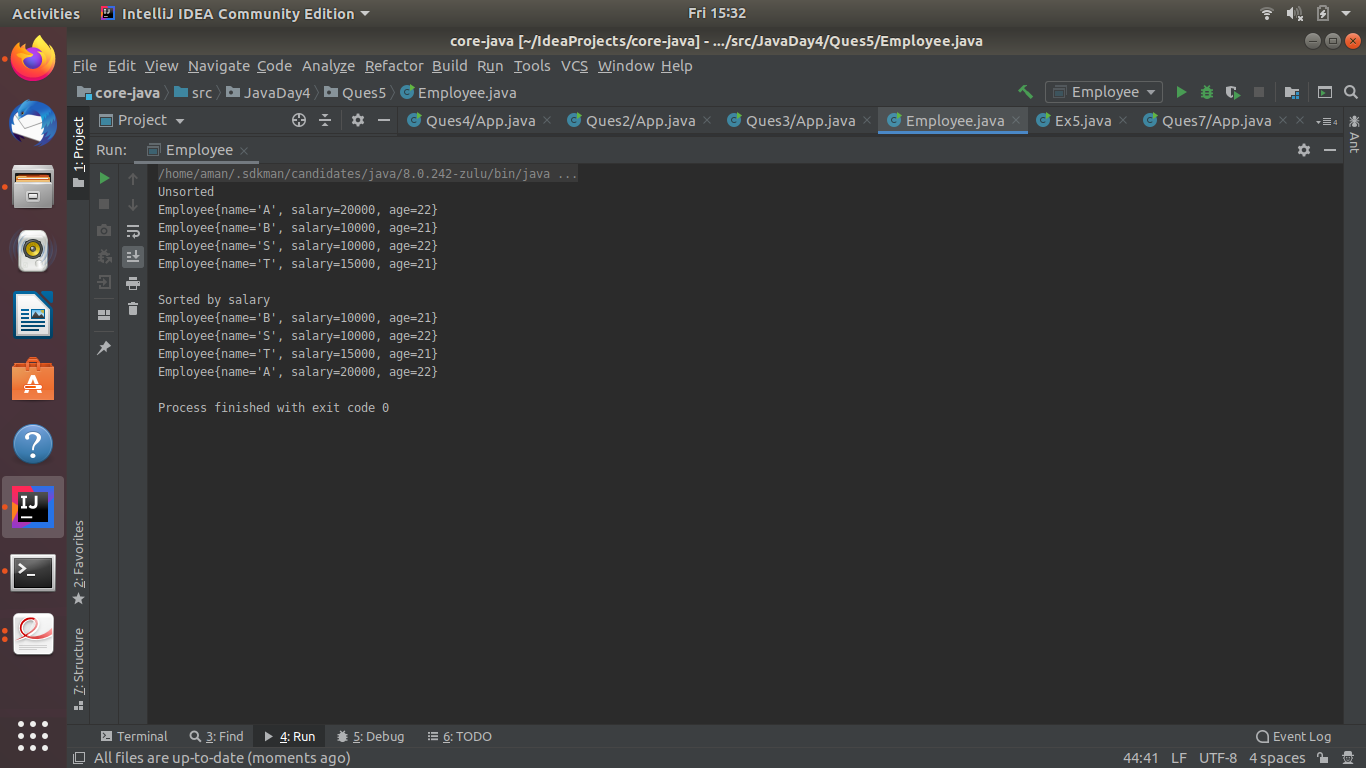
****

1. Write a program to sort Employee objects based on highest salary using Comparator. Employee class{ Int Age; Int Salary; String Name}

**Sol 5.**

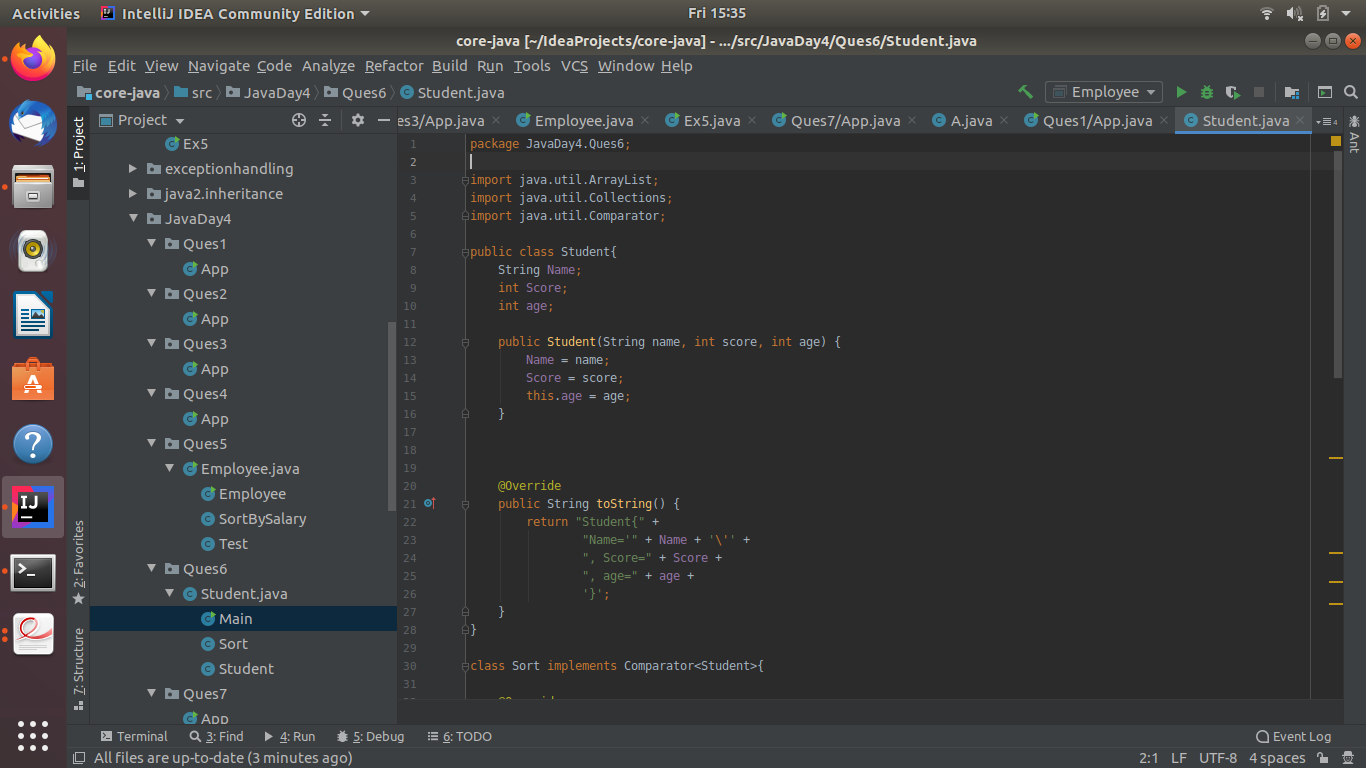
****

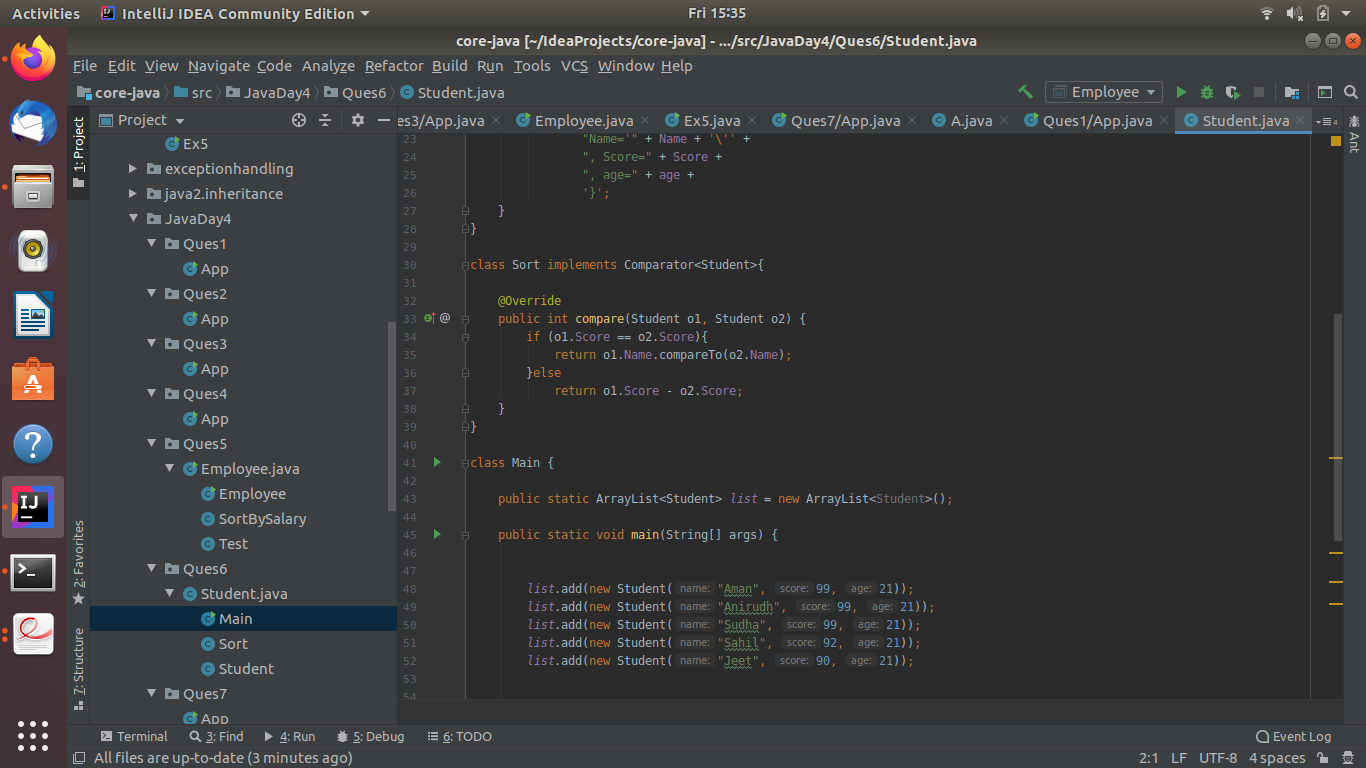
****

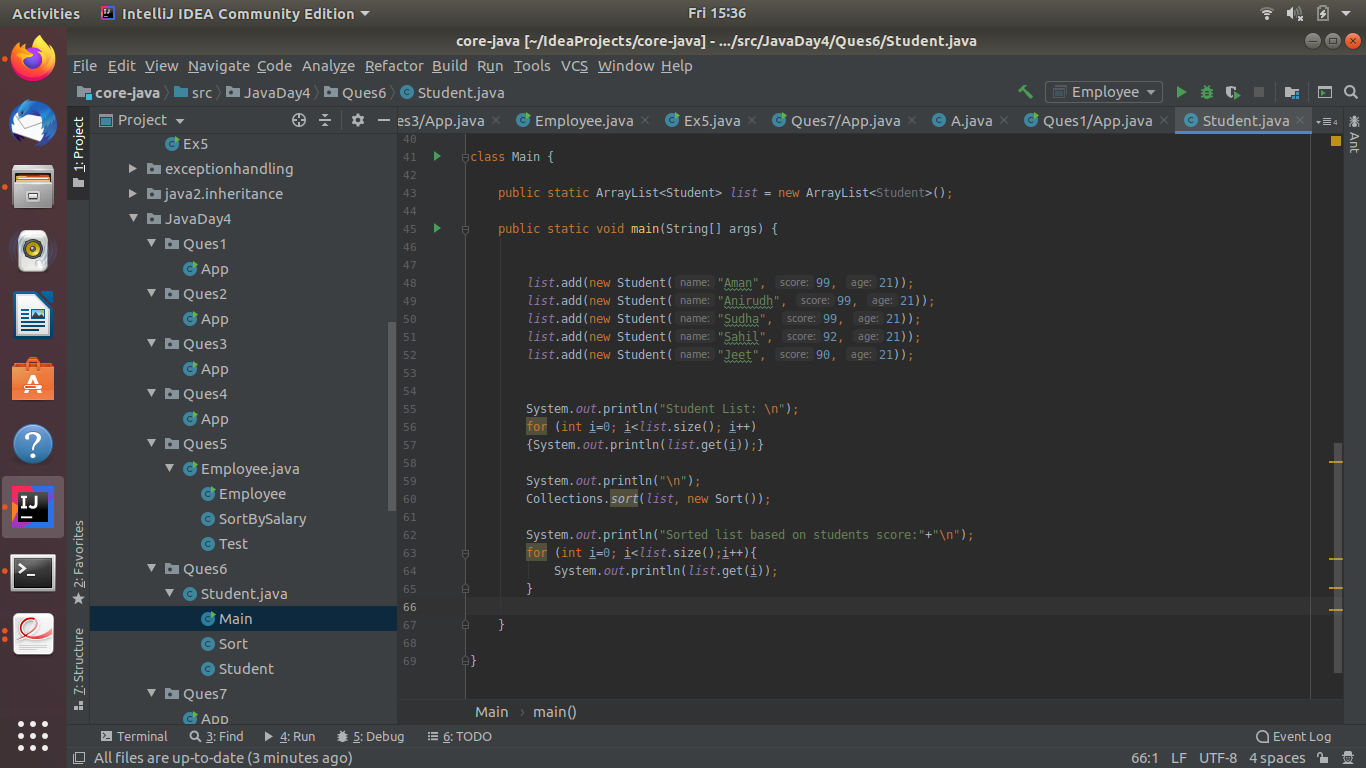
****

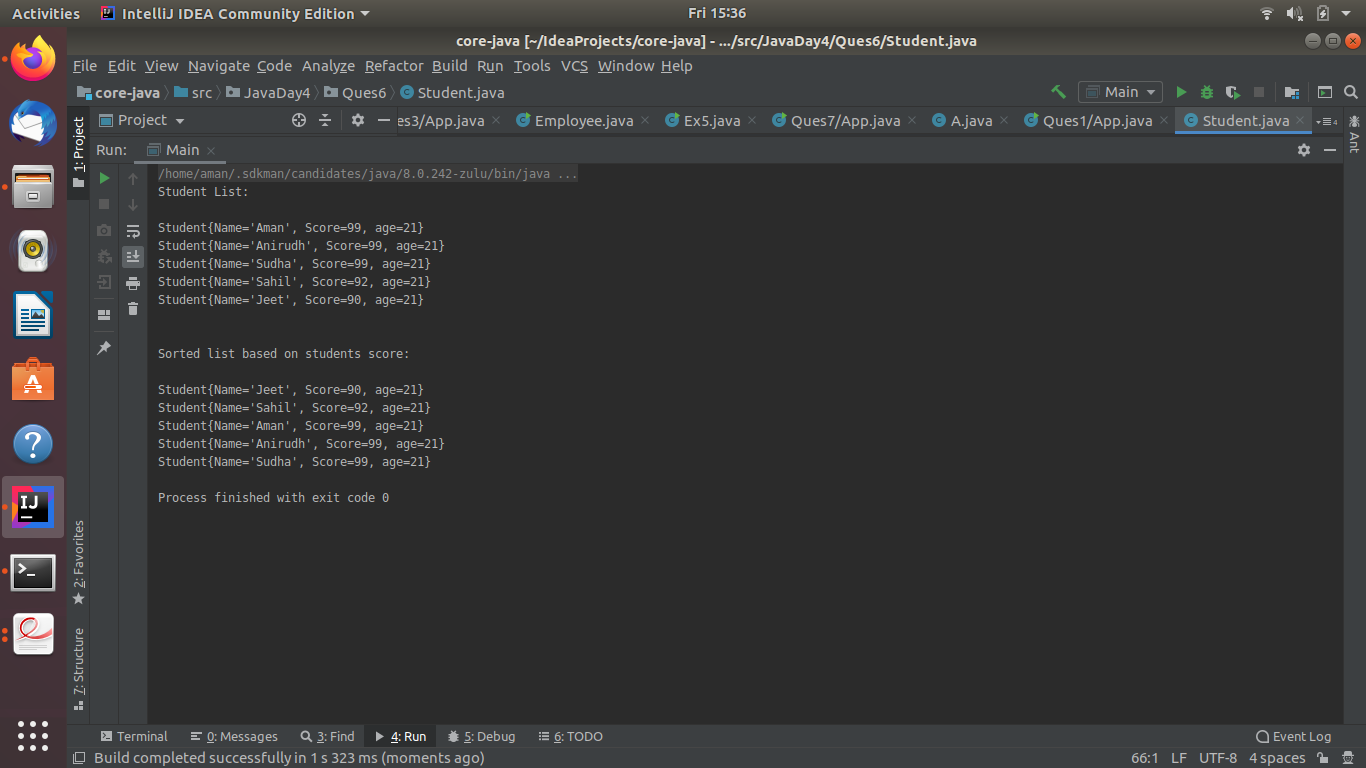
1. Write a program to sort the Student objects based on Score , if the score are same then sort on First Name . Class Student{ String Name; Int Score; Int Age}

**Sol 6.**

****

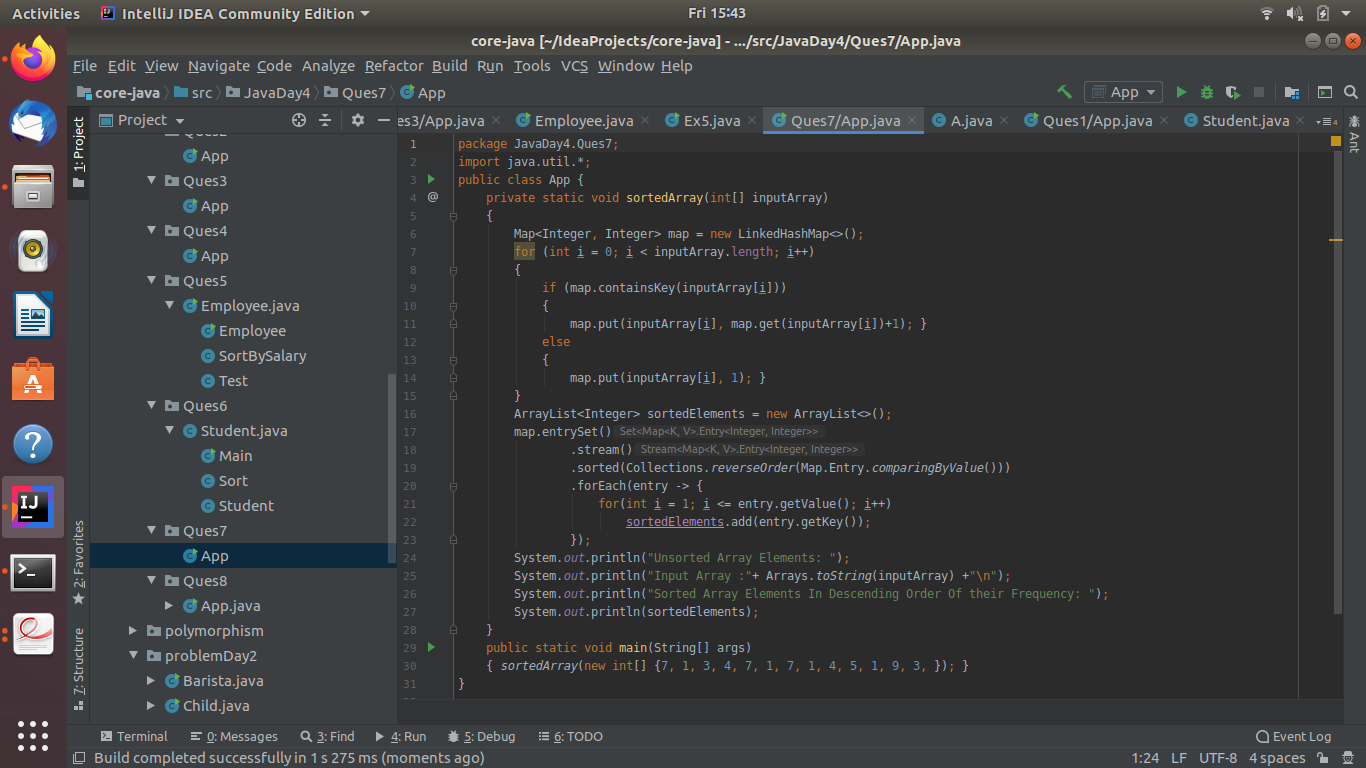
****

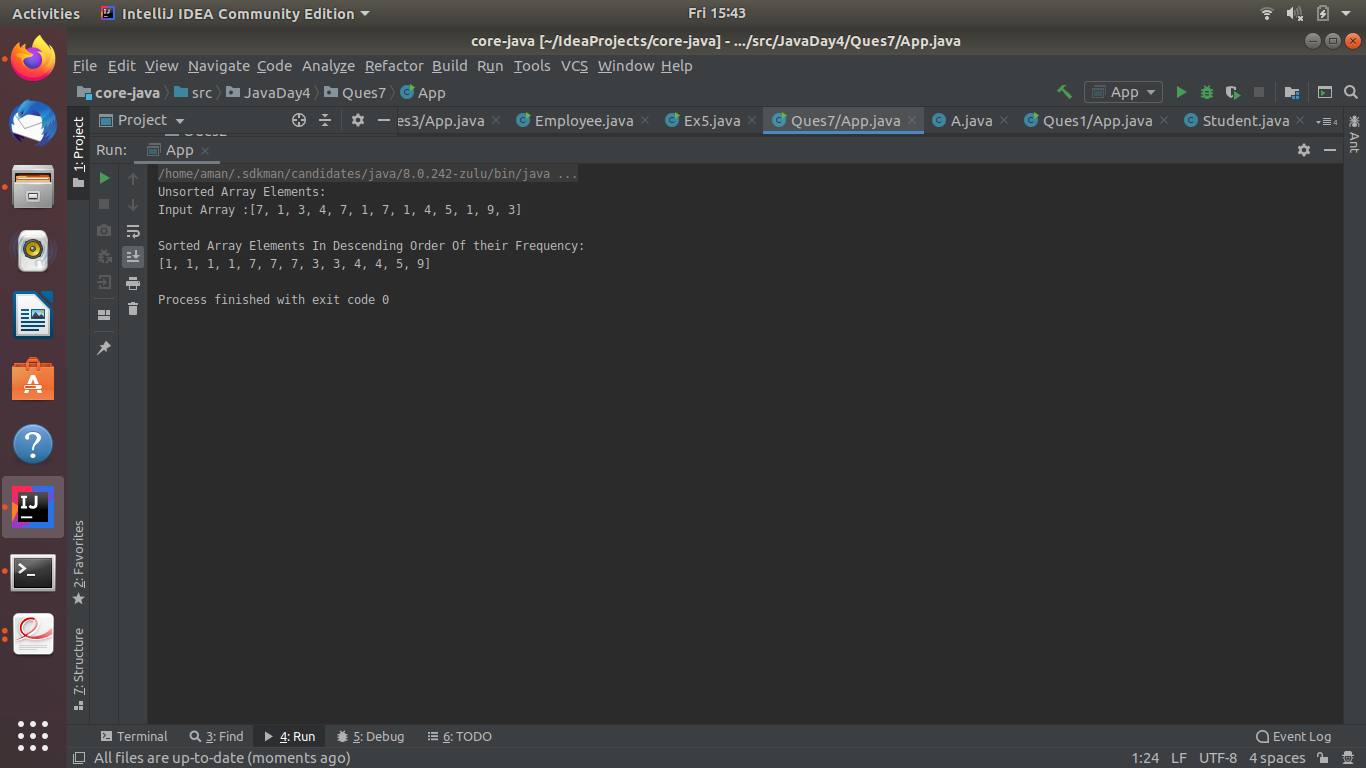
****

****

1. Print the elements of an array in the decreasing frequency if 2 numbers have same frequency then print the one which came first.

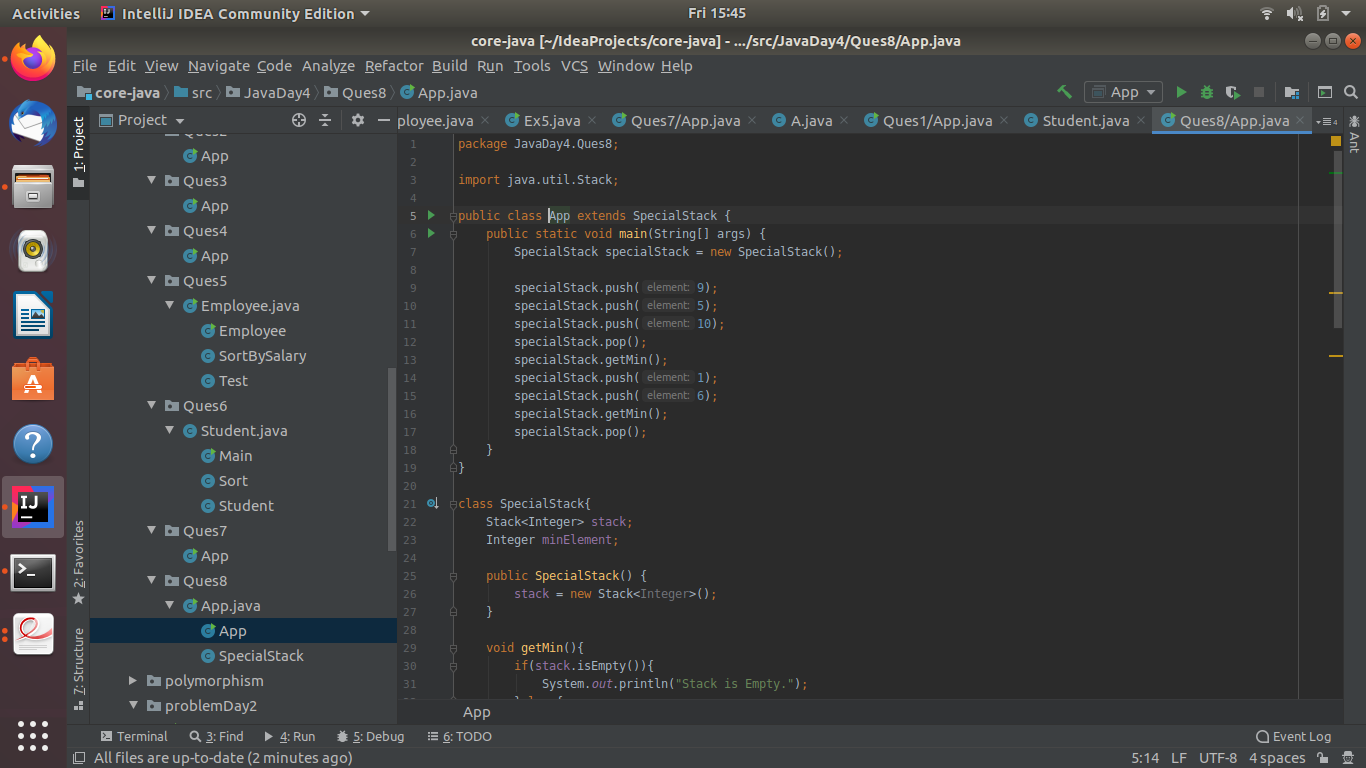
**Sol 7.**

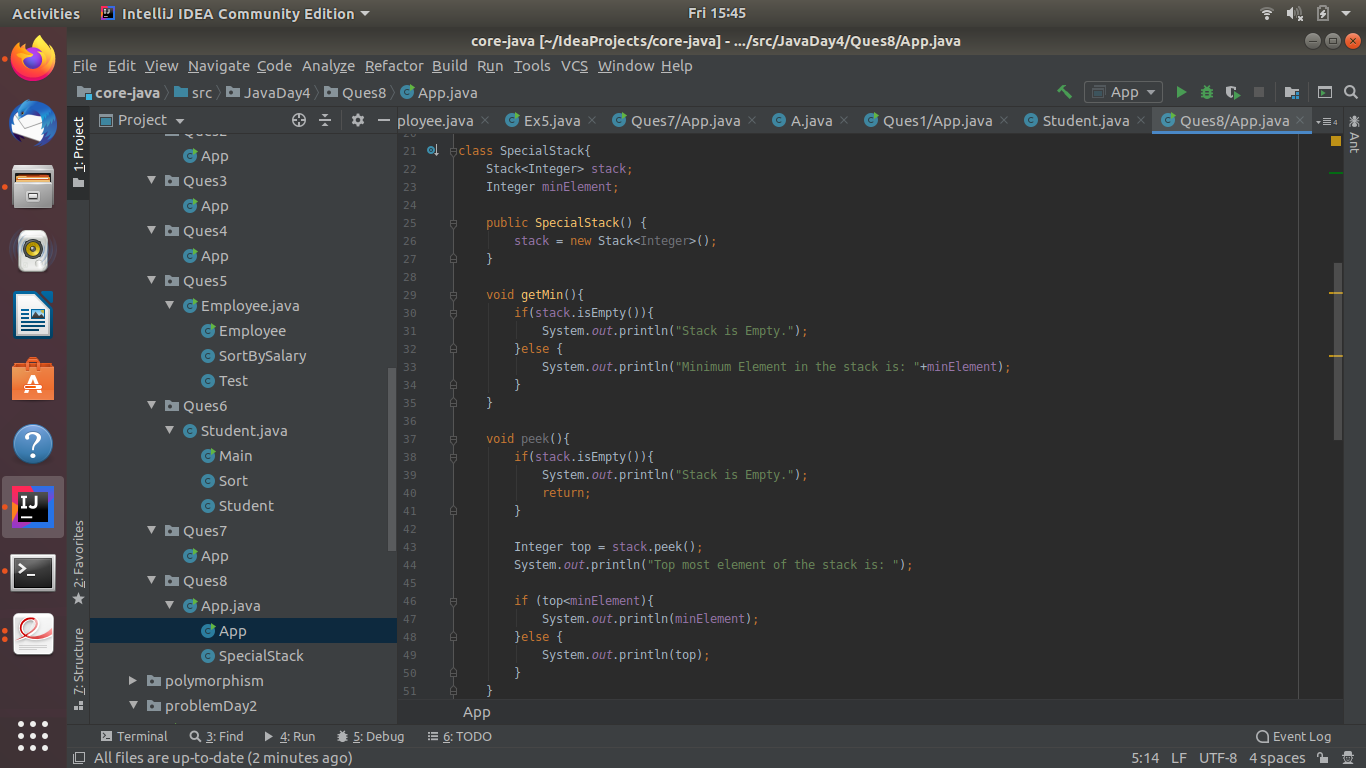
****

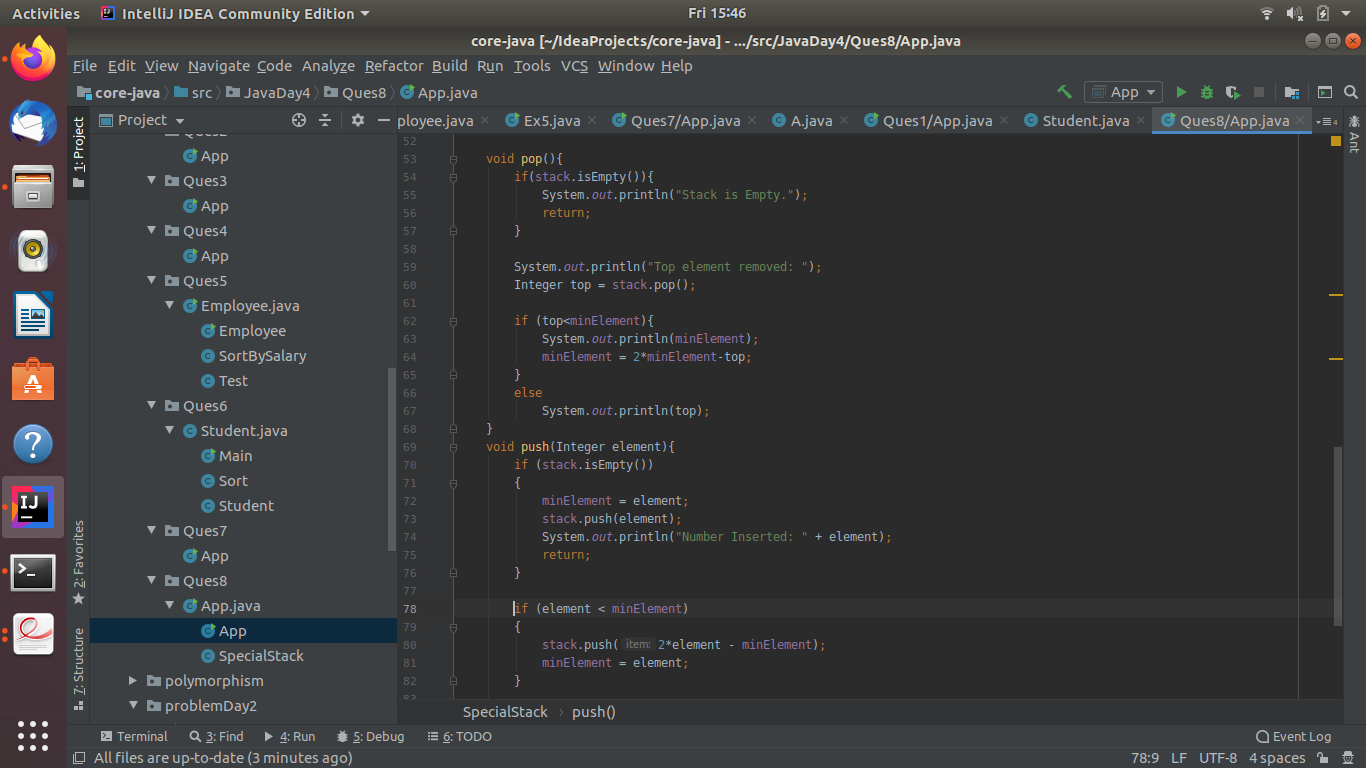
****

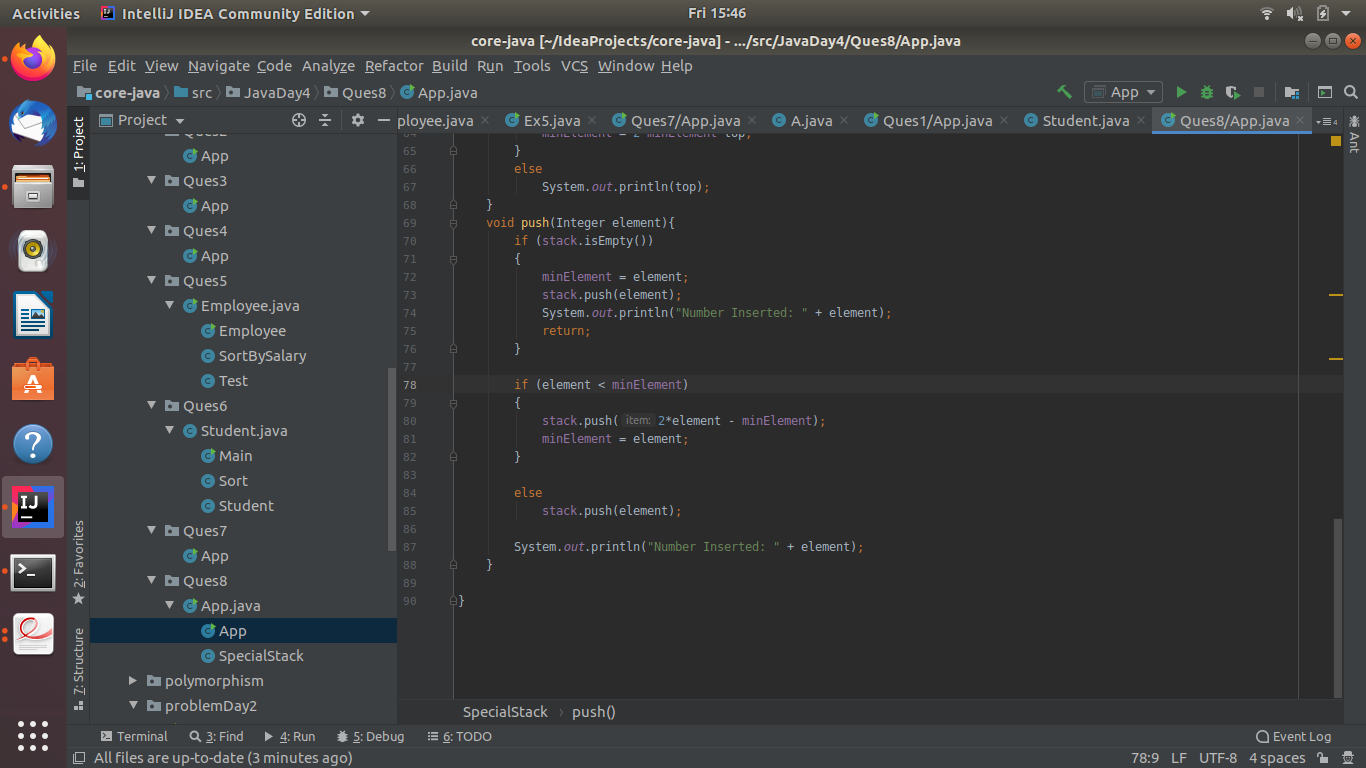
1. Design a Data Structure SpecialStack that supports all the stack operations like push(), pop(), isEmpty(), isFull() and an additional operation getMin() which should return minimum element from the SpecialStack. (Expected complexity ­ O(1))

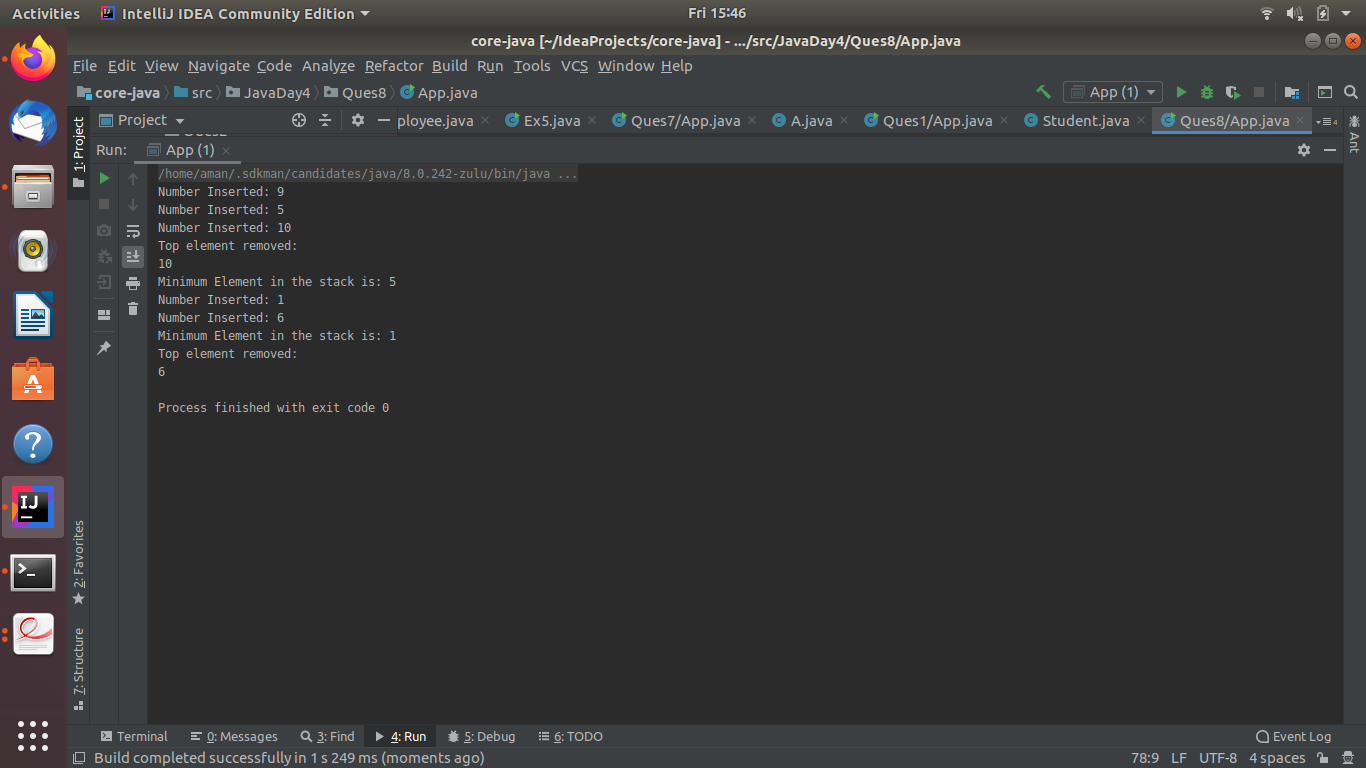
**Sol 8.**

****

****

****

****

****