

1. sort the string in ascending or descending order

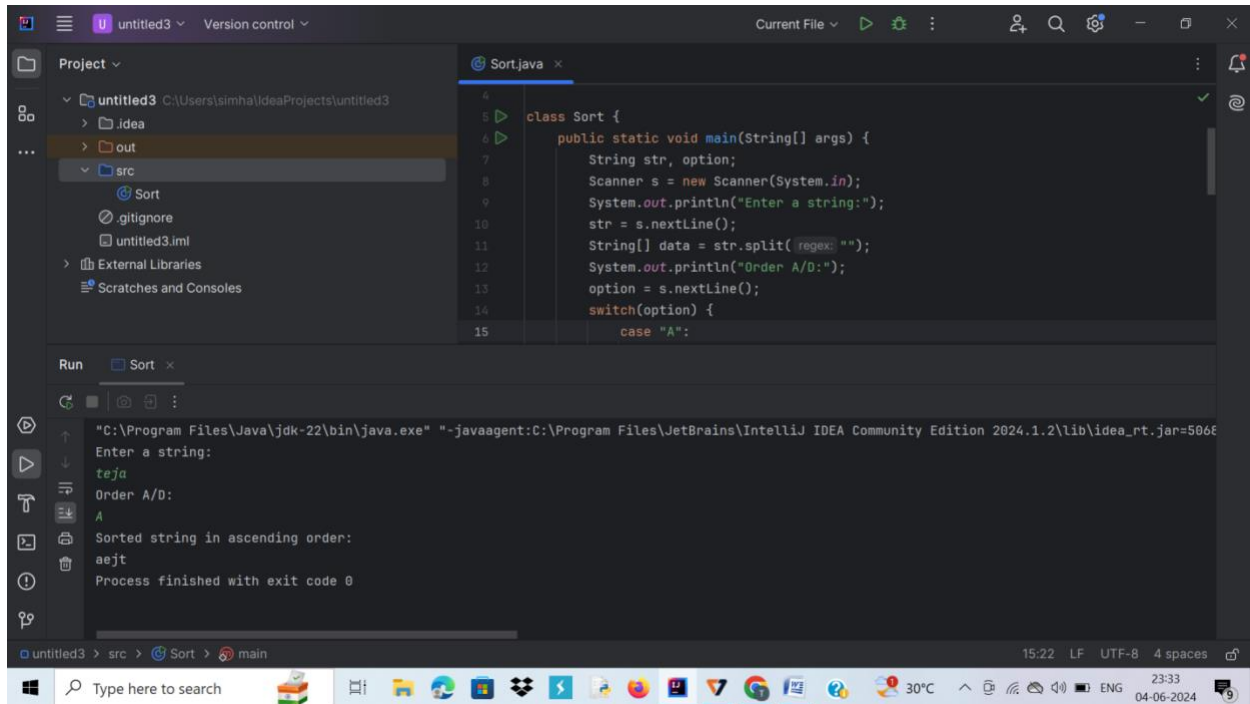
Program:

1. Write a java program to sort the string in ascending or descending order. Choice get from user.

Program:-

```
import java.util. arrays ;
class sort
{
    public static void main ( string[] args )
    {
        string str, option;
        Scanner s = new Scanner ( System.in )
        System.out.println ( "Enter a string:" );
        str = s.nextLine();
        String[] data = str.split(" ");
        System.out.println ( " Order A/D:" );
        Option = s.nextLine();
        switch (option)
        {
            case "A":
                Arrays.sort (data);
                for (string output. data)
            case "D":
                Arrays.sort (data, collection: reverse Order());
                for (string output. data)
            default:
                System.out.println ( "choose the correct option" );
        }
    }
}
```

output:



The screenshot displays the IntelliJ IDEA IDE interface. The top toolbar shows the 'Run' button (a green play icon). The left sidebar contains the 'Project' view, showing a directory structure with 'src' containing 'Sort.java'. The main editor window displays the code for 'Sort.java'.

```
4  
5 class Sort {  
6     public static void main(String[] args) {  
7         String str, option;  
8         Scanner s = new Scanner(System.in);  
9         System.out.println("Enter a string:");  
10        str = s.nextLine();  
11        String[] data = str.split(" ");  
12        System.out.println("Order A/D:");  
13        option = s.nextLine();  
14        switch(option) {  
15            case "A":
```

Below the editor, the 'Run' window shows the execution output for 'Sort'.

```
"C:\Program Files\Java\jdk-22\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.1.2\lib\idea_rt.jar=506f  
Enter a string:  
teja  
Order A/D:  
A  
Sorted string in ascending order:  
aejt  
Process finished with exit code 0
```

The bottom status bar indicates the file is 'untitled3' in the 'src' directory, editing 'Sort.java' at line 15, column 15. The system tray at the bottom shows the time as 15:22, date as 04-06-2024, and temperature as 30°C.

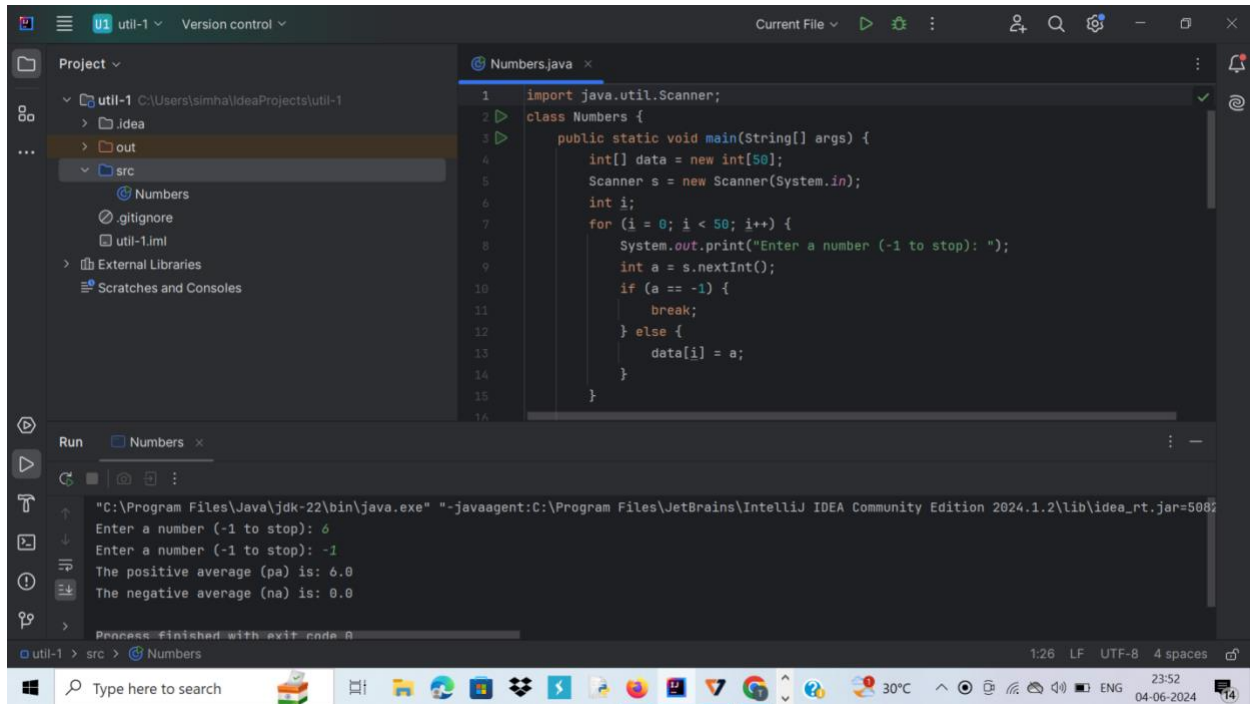
2.

2. Write a java program to read the numbers until -1 encountered. Find the average of positive numbers and negative numbers. Entered by user.

Class numbers

```
{
    public static void main(String args[])
    {
        int[] data = new int[50];
        Scanner s = new Scanner(System.in);
        for(i=0; i<50; i++)
        {
            int a;
            a = s.nextInt();
            a--;
            if(a == -1)
            {
                break;
            }
            else
            {
                data[i] = a;
            }
        }
        int pc=0, ps=0, pa=0, nc=0, ns=0, na=0.
        for(i=0; i<data.length(); i++)
        {
            if(data[i] < 0)
            {
                ns = ns + data[i];
                nc++;
            }
            else
            {
                ps = ps + data[i];
                pc++;
            }
            pa = ps/pc;
            na = ns/nc;
        }
    }
}
```

Output:



```
1 import java.util.Scanner;
2 class Numbers {
3     public static void main(String[] args) {
4         int[] data = new int[50];
5         Scanner s = new Scanner(System.in);
6         int i;
7         for (i = 0; i < 50; i++) {
8             System.out.print("Enter a number (-1 to stop): ");
9             int a = s.nextInt();
10            if (a == -1) {
11                break;
12            } else {
13                data[i] = a;
14            }
15        }
16    }
17 }
```

Run Numbers x

```
"C:\Program Files\Java\jdk-22\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.1.2\lib\idea_rt.jar=5082"
Enter a number (-1 to stop): 6
Enter a number (-1 to stop): -1
The positive average (pa) is: 6.0
The negative average (na) is: 0.0
Process finished with exit code 0
```

util-1 > src > Numbers 1:26 LF UTF-8 4 spaces

3.

```
System.out.println("The pa" + pa);
```

```
System.out.println("The na" + na);
```

3. Write a java program to read a character until * is encountered also count the number of up case and lower case and Numbers entered by the user.

```
class numbers
```

```
{  
    public static void main (String args[])
```

```
{  
    char[] data = new char[25];
```

```
    String str;
```

```
    int up, lp, n;
```

```
    for (i = 0; i < 25; i++)
```

```
{  
        str = s.nextLine();
```

```
        str = str.trim();
```

```
        if (str.equals("*"))
```

```
{  
            break;
```

```
        }  
        else
```

```
{  
            data[i] = str;
```

```
        }  
    }  
    for (i = 0; i < str.length() - 1; i++)
```

```
{  
        if (str.charAt(i) >= 'a' && str.charAt(i) <= 'z')
```

```
{  
            up = up + 1;
```

Output:

The screenshot displays the Programiz Online Java Compiler interface. At the top, there's a navigation bar with the Programiz logo, a 'Premium Coding Courses by Programiz' banner, and a 'Learn More' button. Below this, the main workspace is divided into three sections: a file explorer on the left showing 'Main.java', a code editor in the center, and an output console on the right. The code editor contains a Java program that processes the string 'Saveetha School of engineering'. The program counts the number of lowercase letters, uppercase letters, and digits. The output console shows the execution results: 'Lower case: 31', 'Upper case: 2', and 'Numbers: 0'. The status bar at the bottom indicates the code execution was successful.

```
1 if (str.equals("")) {
2     break;
3 }
4 for (int i = 0; i < str.length(); i++) {
5     char ch = str.charAt(i);
6     if (ch >= 'a' && ch <= 'z') {
7         lp=lp+1;
8     } else if (ch >= 'A' && ch <= 'Z') {
9         up=up+1;
10    } else if (ch >= '0' && ch <= '9') {
11        n=n+1;
12    }
13 }
14 System.out.println("Lower case: " + lp);
15 System.out.println("Upper case: " + up);
16 System.out.println("Numbers: " + n);
17 }
```

Output:

```
java -cp /tmp/vev2NuqblS/Numbers
Saveetha School of engineering
Teja
Student
engineering

Teja
Student

*
Lower case: 31
Upper case: 2
Numbers: 0

=== Code Execution Successful ===
```

4.


```

else if (str.charAt(i) >= 'A' && str.charAt(i) <= 'Z')
{
    lP = lP + 1;
}
else
{
    n = n + 1;
}
}
System.out.println("lower case:" + lP);
System.out.println("upper case:" + uP);
System.out.println("numbers:" + n);
}
}

```

4. Write a java program to find m^{th} maximum number and n^{th} minimum number in an array and find the sum and difference of it.

Class Sorting

```

{
    public static void main (String args[])
    {
        int n, m, size, sum = 0, diff = 0;
        Scanner s = new Scanner (System.in);
        System.out.println("Enter the size of the array:");
        size = 10;
        int [] data = new int [size];
        for (i = 0; i < size; i++)
        {
            data[i] = s.nextInt();
        }
        for (i = 0; i < size; i++)

```



```

    {
        for (j=i+1; j<size; j++)
        {
            int temp=0;
            if (data[i]>data[j])
            {
                temp=data[i];
                data[i]=data[j];
                data[j]=temp;
            }
        }
    }
    Sum=data[(size-1)-(m-1)]+data[n-1];
    diff=data[(size-1)-(m-1)]-data[n-1];
    System.out.println("sum:"+sum);
    System.out.println("diff:"+diff);
}
}

```

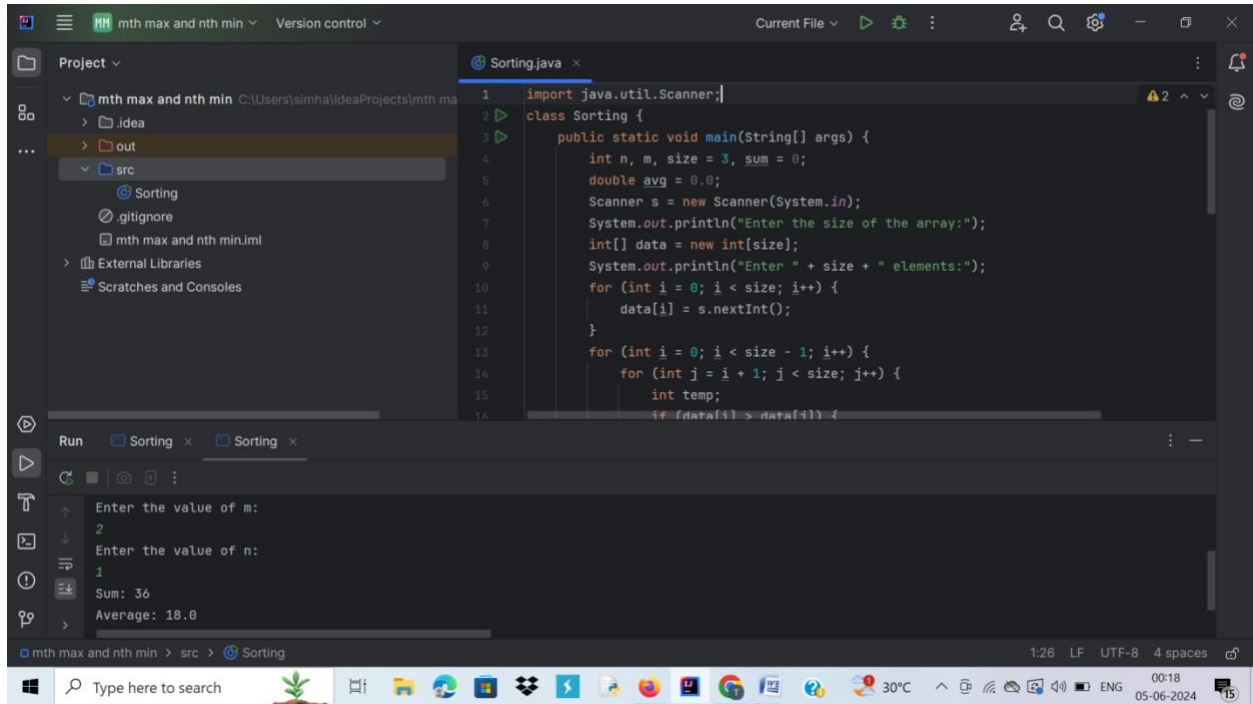
5. Bank is the class that provides method to get rate of interest. but rate of interest may differ according to banks. SBI is providing 8.4%, ICICI provides 7.3 percentage and Axis banks 9.7 % of rate of interest. Write a java program for above scenario.

```

class bank
{
    double get ROI();
}
class SBI extends bank
{
    double get ROI()

```

Output:



```
1 import java.util.Scanner;
2 class Sorting {
3     public static void main(String[] args) {
4         int n, m, size = 3, sum = 0;
5         double avg = 0.0;
6         Scanner s = new Scanner(System.in);
7         System.out.println("Enter the size of the array:");
8         int[] data = new int[size];
9         System.out.println("Enter " + size + " elements:");
10        for (int i = 0; i < size; i++) {
11            data[i] = s.nextInt();
12        }
13        for (int i = 0; i < size - 1; i++) {
14            for (int j = i + 1; j < size; j++) {
15                int temp;
16                if (data[i] > data[j]) {
```

Run Sorting x Sorting x

Enter the value of m:
2
Enter the value of n:
1
Sum: 36
Average: 18.0

1:26 LF UTF-8 4 spaces

5.

```

    {
        for (j=i+1; j<size; j++)
        {
            int temp=0;
            if (data[i]>data[j])
            {
                temp=data[i];
                data[i]=data[j];
                data[j]=temp;
            }
        }
    }
    Sum=data[(size-1)-(m-1)]+data[n-1];
    diff=data[(size-1)-(m-1)]-data[n-1];
    System.out.println("sum:"+sum);
    System.out.println("diff:"+diff);
}
}

```

5. Bank is the class that provides method to get rate of interest. but rate of interest may differ according to banks. SBI is providing 8.4%, ICICI provides 7.3 percentage and Axis banks 9.7 % of rate of interest. Write a java program for above scenario.

```

class bank
{
    double get ROI();
}
class SBI extends bank
{
    double get ROI()

```

```

    {
        return 8.4;
    }
}

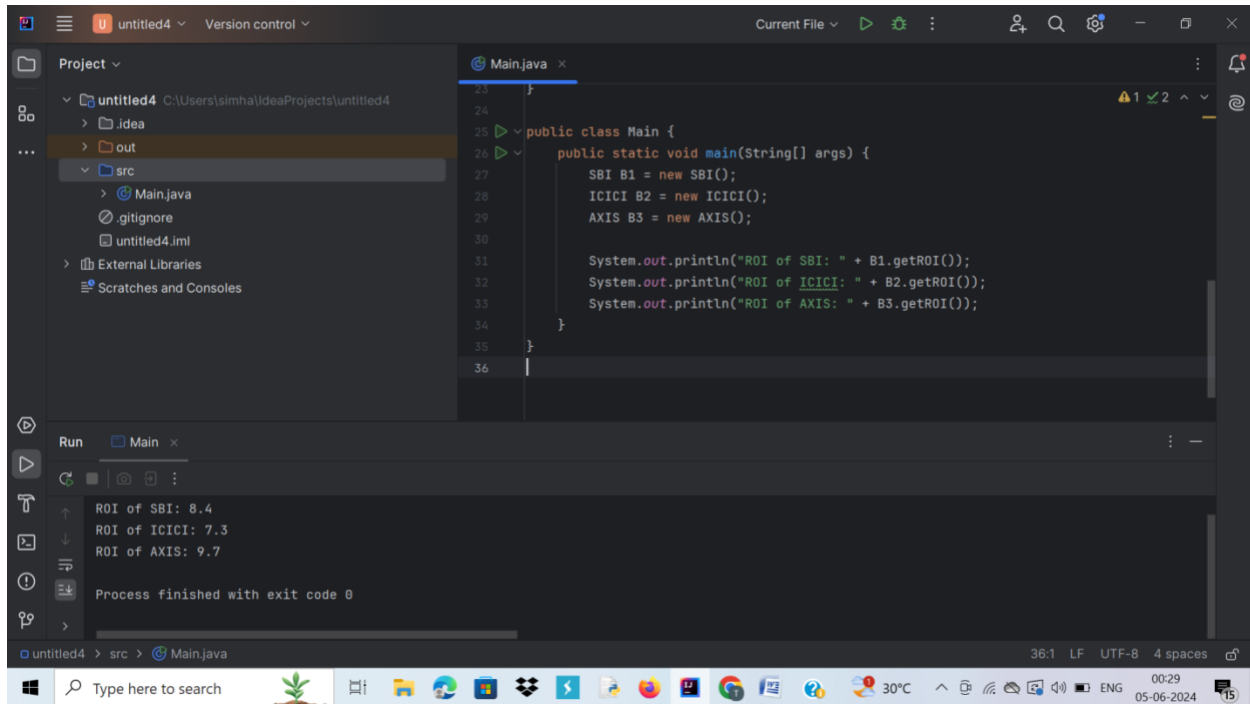
Class ICICI extends bank
{
    double get ROI()
    {
        return 7.3;
    }
}

Class Axis extends bank
{
    double get ROI()
    {
        return 9.7;
    }
}

Class main
{
    public static void main (String args[])
    {
        SBI B1 = new SBI();
        ICICI B2 = new ICICI();
        Axis B3 = new Axis();
        System.out.println("ROI SBI" + B1.get ROI());
        System.out.println("ROI ICICI" + B2.get ICICI());
        System.out.println("ROI Axis" + B3.get Axis());
    }
}

```

Output:



The screenshot shows an IDE window with a project named 'untitled4'. The project structure includes a 'src' folder containing 'Main.java'. The code in 'Main.java' defines a 'Main' class with a 'main' method that creates instances of 'SBI', 'ICICI', and 'AXIS' classes and prints their 'ROI' values. The output console shows the results of the program execution: 'ROI of SBI: 8.4', 'ROI of ICICI: 7.3', and 'ROI of AXIS: 9.7'. The process finished with exit code 0.

```
23 }
24
25 public class Main {
26     public static void main(String[] args) {
27         SBI B1 = new SBI();
28         ICICI B2 = new ICICI();
29         AXIS B3 = new AXIS();
30
31         System.out.println("ROI of SBI: " + B1.getROI());
32         System.out.println("ROI of ICICI: " + B2.getROI());
33         System.out.println("ROI of AXIS: " + B3.getROI());
34     }
35 }
36
```

Run Main x

ROI of SBI: 8.4
ROI of ICICI: 7.3
ROI of AXIS: 9.7

Process finished with exit code 0

untitled4 > src > Main.java 36:1 LF UTF-8 4 spaces

6. Implement or Write a Java program for student total using constructor.

```
Class student
```

```
{
```

```
    int m1, m2, tot;
```

```
    Student (int mark1, int mark2)
```

```
{
```

```
        m1 = mark1
```

```
        m2 = mark2
```

```
}
```

```
class tot extends student
```

```
{
```

```
    tot = m1 + m2
```

```
    void display()
```

```
{
```

```
        System.out.println("Total: " + tot);
```

```
}
```

```
}
```

```
Class main
```

```
{
```

```
    public static void main (String args[])
```

```
{
```

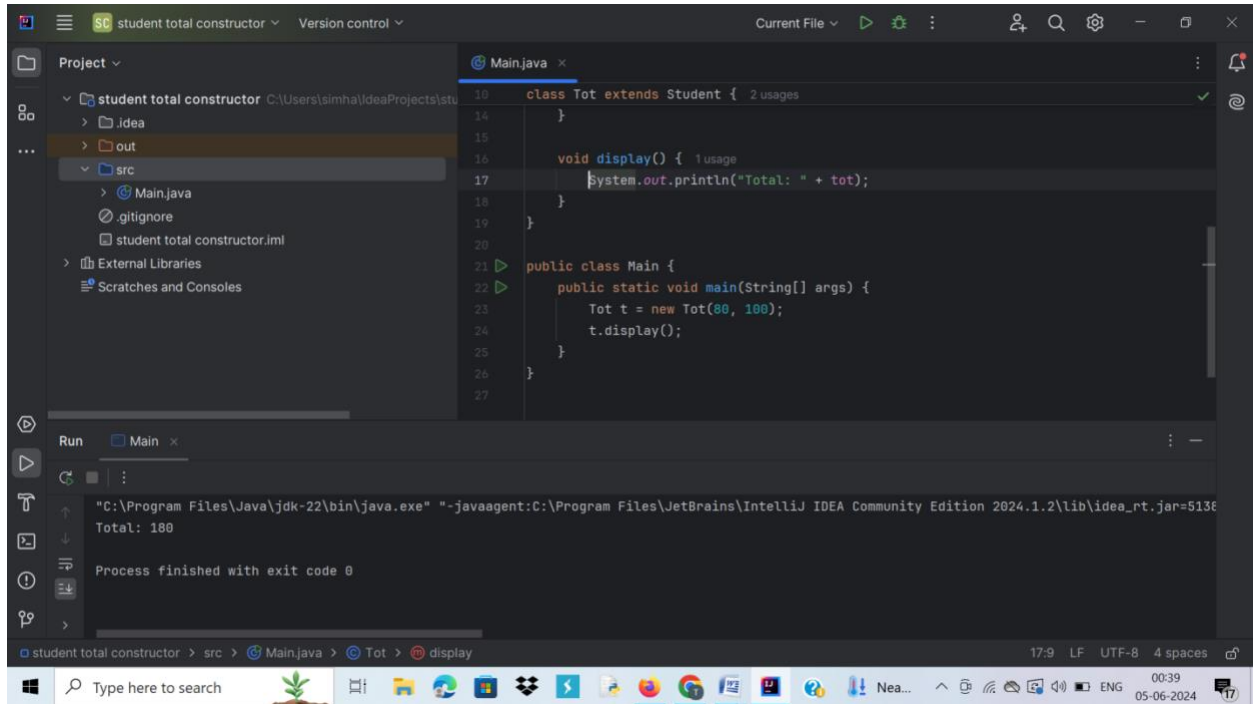
```
        tot t (80, 100);
```

```
        t.display();
```

```
}
```

```
}
```


Output:



The screenshot displays the IntelliJ IDEA IDE interface. The top toolbar includes icons for running and debugging. The left sidebar shows the project structure for 'student total constructor', with the 'src' directory expanded to show 'Main.java'. The main editor window displays the following Java code:

```
10 class Tot extends Student { 2 usages
14 }
15
16 void display() { 1 usage
17     System.out.println("Total: " + tot);
18 }
19
20
21 public class Main {
22     public static void main(String[] args) {
23         Tot t = new Tot(80, 100);
24         t.display();
25     }
26 }
27
```

Below the code editor, the 'Run' tab is active, showing the execution command and output:

```
"C:\Program Files\Java\jdk-22\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.1.2\lib\idea_rt.jar=5136
Total: 180
Process finished with exit code 0
```

The bottom status bar indicates the file path 'student total constructor > src > Main.java > Tot > display', the cursor position '17:9', and encoding 'UTF-8'.

7.

7. Write a java program which converts roman numerals into integers

```
import java.util.*;
```

```
class roman
```

```
{  
    public static void main (String args[])
```

```
{  
    String str;
```

```
    int length, final;
```

```
    Scanner s = new Scanner (System.in);
```

```
    System.out.println ("Enter roman numerals:");
```

```
    str = s.nextLine();
```

```
    final = 0;
```

```
    length = str.length();
```

```
    for (int i=0; i<length; i++)
```

```
    {  
        char ch = str.charAt(i);
```

```
        char next ch = str.charAt(i+1);
```

```
        if (ch == 'M')
```

```
        {  
            final += 1000;
```

```
        }
```

```
        else if (ch == 'C')
```

```
        {  
            if (next ch == 'M')
```

```
            {  
                final += 900;
```

```
                i++;
```

```
            }  
            else if (next ch == 'D')
```

```
            {
```

```
                System.out.println ("Invalid roman numerical");
```

```
            }
```

```
            System.out.println ("Result" + final);
```

Output:

Online Java Compiler

programiz.com/java-programming/online-compiler/

Programiz Online Java Compiler

Run

```
1- import java.util.Scanner;
2
3- class Roman {
4-     public static void main(String[] args) {
5         String str;
6         int length, finalValue;
7         Scanner s = new Scanner(System.in);
8         System.out.println("Enter Roman numerals:");
9         str = s.nextLine().toUpperCase(); // Convert to
           uppercase to handle case sensitivity
10        finalValue = 0;
11        length = str.length();
12
13-        for (int i = 0; i < length; i++) {
14            char ch = str.charAt(i);
15            int value = getValue(ch);
16
17-            if (i + 1 < length) {
18                char nextCh = str.charAt(i + 1);
19                int nextValue = getValue(nextCh);
20            }
21        }
22    }
23}
```

Output

```
java -cp /tmp/tir7EVzorA/Roman
Enter Roman numerals:
teja
Invalid Roman numeral: T
Invalid Roman numeral: E
Invalid Roman numeral: E
Invalid Roman numeral: J
Invalid Roman numeral: J
Invalid Roman numeral: A
Invalid Roman numeral: A
Result: 0

=== Code Execution Successful ===
```

Online Java Compiler

programiz.com/java-programming/online-compiler/

Programiz Online Java Compiler

Run

```
1- import java.util.Scanner;
2- class Roman {
3-     public static void main(String[] args) {
4         String str;
5         int length, finalValue;
6         Scanner s = new Scanner(System.in);
7         System.out.println("Enter Roman numerals:");
8         str = s.nextLine().toUpperCase();
9         finalValue = 0;
10        length = str.length();
11-        for (int i = 0; i < length; i++) {
12            char ch = str.charAt(i);
13            int value = getValue(ch);
14-            if (i + 1 < length) {
15                char nextCh = str.charAt(i + 1);
16                int nextValue = getValue(nextCh);
17-                if (value < nextValue) {
18                    finalValue -= value;
19                } else {
20                    finalValue += value;
21                }
22            }
23        }
24    }
25}
```

Output

```
java -cp /tmp/rxmXV9I9Zz/Roman
Enter Roman numerals:
mcn
Invalid Roman numeral: N
Invalid Roman numeral: N
Result: 1100

=== Code Execution Successful ===
```