



ASSIGNMENT

Module 2 : Object Oriented Programming in JAVA

ABSTRACT

In this assignment, I have completed 23 questions on java basic topics like decision making statement, loop and arrays. Every question starts from new page along with its code and output.

sarang deodhar

220350320026

Q1. Design a Java program to get a number from user and find the given number is positive or negative. Display the message.

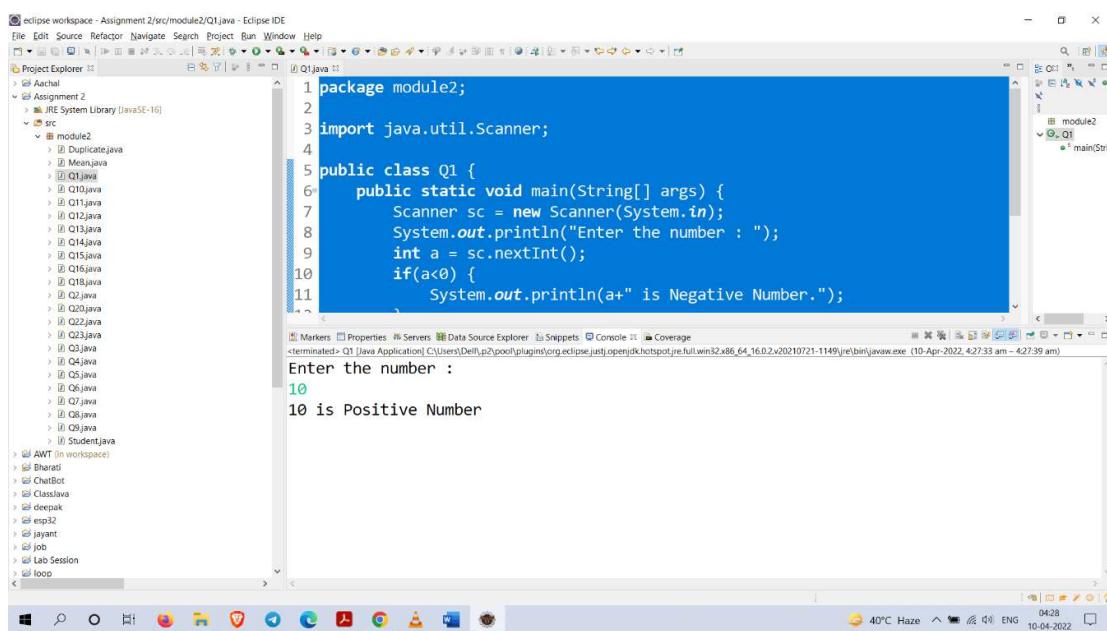
Code :

```
package module2;

import java.util.Scanner;

public class Q1 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the number : ");
        int a = sc.nextInt();
        if(a<0) {
            System.out.println(a+" is Negative Number.");
        }
        else if(a>0) {
            System.out.println(a+" is Positive Number");
        }
        sc.close();
    }
}
```

Output :



Q2. Write a Java program to calculate a Factorial of a number

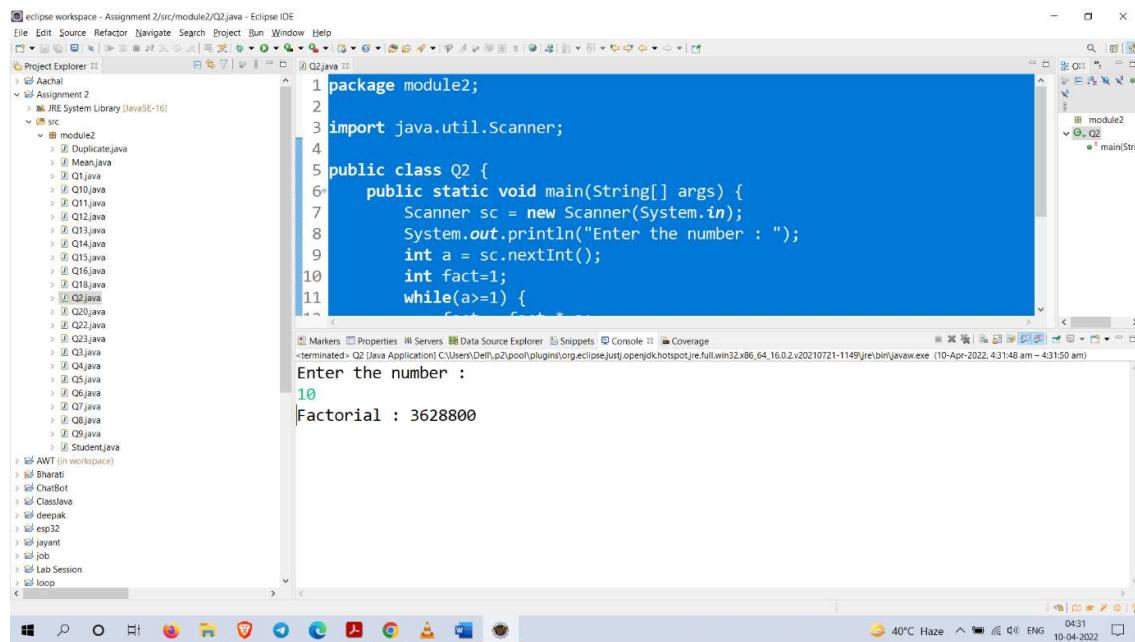
Code :

```
package module2;

import java.util.Scanner;

public class Q2 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the number : ");
        int a = sc.nextInt();
        int fact=1;
        while(a>=1) {
            fact = fact * a;
            a--;
        }
        System.out.println("Factorial : "+fact);
        sc.close();
    }
}
```

Output :



Q3. Write a Java program to get the age of a person and find the age group of that person as

Age	Category
0-3	Toddlers
4-8	Kids
9-12	Child
13-18	Teens
19 – 40	Adults
41- 60	Matured Adults
61 – 75	Seniors
>76	Super Seniors

Code :

```
package module2;

import java.util.Scanner;

public class Q3 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the number : ");
        int a = sc.nextInt();
        if(a<=3) {
            System.out.println("Toddlers");
        }else if(a<=8) {
            System.out.println("Kids");
        }else if(a<=12) {
            System.out.println("Child");
        }else if(a<=18) {
            System.out.println("Teens");
        }else if(a<=40) {
            System.out.println("Adults");
        }else if(a<=60) {
            System.out.println("Matured Adults");
        }else if(a<=75) {
            System.out.println("Seniors");
        }else if(a>76) {
            System.out.println("Super Seniors");
        }
        sc.close();
    }
}
```

Output :

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure with modules Aachal, Assignment 2, and src. Under src, there are several Java files: Q1.java, Q10.java, Q11.java, Q12.java, Q13.java, Q14.java, Q15.java, Q16.java, Q2.java, Q20.java, Q3.java, Q4.java, Q5.java, Q6.java, Q7.java, Q8.java, Q9.java, and Student.java.
- Code Editor:** Displays the contents of Q3.java:

```
1 package module2;
2
3 import java.util.Scanner;
4
5 public class Q3 {
6     public static void main(String[] args) {
7         Scanner sc = new Scanner(System.in);
8         System.out.println("Enter the number : ");
9         int a = sc.nextInt();
10        if(a<=3) {
11            System.out.println("Toddlers");
12        }
13        else if(a<=10) {
14            System.out.println("Adults");
15        }
16        else {
17            System.out.println("Older");
18        }
19    }
20}
```
- Console:** Shows the output of the program execution:

```
Enter the number :
24
Adults
```
- System Tray:** Shows the system status including temperature (40°C), battery level (Haze), and date/time (10-Apr-2022, 04:38).

Q4. Write a program called CozaLozaWoza which prints the numbers 1 to 110, 11 numbers per line. The program shall print "Coza" in place of the numbers which are multiples of 3, "Loza" for multiples of 5, "Woza" for multiples of 7, "CozaLoza" for multiples of 3 and 5, and so on.

The output shall look like:

```
1 2 Coza 4 Loza Coza Woza 8 Coza Loza 11
Coza 13 Woza CozaLoza 16 17 Coza 19 Loza CozaWoza 22
23 oza Loza 26 Coza Woza 29 CozaLoza 31 32 Coza
```

Code :

```
package module2;

public class Q4 {
    public static void main(String[] args) {
        for(int i=1;i<=110;i++) {

            if(i%3==0 && i%5==0)
                System.out.print("CozaLoza");
            else if(i%3==0)
                System.out.print("Coza");
            else if(i%5==0)
                System.out.print("Loza");
            if(i%7==0)
                System.out.print("Woza");
            if(i%3!=0 && i%5!=0 && i%7!=0)
                System.out.print(i);
            if(i%11==0) {
                System.out.println();
                continue;
            }
            System.out.print(" ");
        }
    }
}
```

Output :

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows a Java project named "Assignment 2" containing several source files (Q1.java, Q11.java, Q12.java, etc.) and a module named "module2".
- Code Editor:** Displays the content of the file "Q4.java". The code prints a sequence of numbers followed by "Coza" or "Woza" based on their remainders when divided by 3 and 5.
- Console:** Shows the terminal output of the program execution, displaying the printed sequence of numbers and strings.
- System Information:** The taskbar at the bottom shows system status like temperature (40°C), battery level (Haze), and network (ENG).

```
3 public class Q4 {  
4     public static void main(String[] args) {  
5         for(int i=1;i<=110;i++) {  
6             if(i%3==0 && i%5==0)  
7                 System.out.print("CozaLoza");  
8             else if(i%3==0)  
9                 System.out.print("Coza");  
10            else if(i%5==0)  
11                System.out.print("Woza");  
12            if(i%7==0)  
13                System.out.print("Coza");  
14        }  
15    }  
16}
```

1 2 Coza 4 Loza Coza Woza 8 Coza Loza 11
Coza 13 Woza CozaLoza 16 17 Coza 19 Loza CozaWoza 22
23 Coza Loza 26 Coza Woza 29 CozaLoza 31 32 Coza
34 LozaWoza Coza 37 38 Coza Loza 41 CozaWoza 43 44
CozaLoza 46 47 Coza Woza Loza Coza 52 53 Coza Loza
Woza Coza 58 59 CozaLoza 61 62 CozaWoza 64 Loza Coza
67 68 Coza LozaWoza 71 Coza 73 74 CozaLoza 76 Woza
Coza 79 Loza Coza 82 83 CozaWoza Loza 86 Coza 88
89 CozaLoza Woza 92 Coza 94 Loza Coza 97 Woza Coza
Loza 101 Coza 103 104 CozaLozaWoza 106 107 Coza 109 Loza

Q5. Write a Java code to get a number and find whether it is prime number or not. (note:Prime number is the number divisible by 1 and itself only)

Code :

```
package module2;

import java.util.Scanner;

public class Q5 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter a number : ");
        int a = sc.nextInt();
        int count = 0;
        for(int i=2;i<=a/2;i++) {
            if(a%i==0) {
                count++;
                break;
            }
        }
        if(count == 0)
            System.out.println(a+" is a prime number.");
        else if(count == 1)
            System.out.println(a+" is a not prime number.");
        sc.close();
    }
}
```

Output :

Roll No.: 220350320026
Name: Sarang G. Deodhar
PG-DAC March 2022
Module 2 : Object Oriented Programming in JAVA

The screenshot shows the Eclipse IDE interface with the following details:

- Title Bar:** eclipse workspace - Assignment 2/src/module2/Q5.java - Eclipse IDE
- Menu Bar:** File Edit Source Refactor Navigate Search Project Run Window Help
- Toolbar:** Standard Eclipse toolbar with various icons.
- Project Explorer:** Shows the project structure with packages like Aachal, Assignment 2, JRE System Library (javaSE-16), and src. Under src, there is a module2 package containing files Q5.java, Duplicate.java, Mean.java, Q1.java, Q10.java, Q11.java, Q12.java, Q13.java, Q14.java, Q15.java, Q16.java, Q18.java, Q2.java, Q20.java, Q22.java, Q23.java, Q3.java, Q4.java, Q5.java, Q6.java, Q7.java, Q8.java, Q9.java, and Student.java.
- Code Editor:** The main window displays the Q5.java file content:

```
1 package module2;
2
3 import java.util.Scanner;
4
5 public class Q5 {
6     public static void main(String[] args) {
7         Scanner sc = new Scanner(System.in);
8         System.out.println("Enter a number : ");
9         int a = sc.nextInt();
10        int count = 0;
11        for(int i=2;i<=a/2;i++) {
12            if(a % i == 0) {
13                count++;
14            }
15        }
16        if(count > 0) {
17            System.out.println(a + " is not a prime number.");
18        } else {
19            System.out.println(a + " is a prime number.");
20        }
21    }
22}
```
- Output Console:** Shows the terminal output:

```
Enter a number :
50
50 is a not prime number.
```
- Bottom Status Bar:** Shows system information: 40°C Haze, 04:42, ENG, 10-04-2022.

Q6. Design a Java program to print the following pattern for the positive value 'n'. Sample Output.

```
Enter any number...
5
*
* *
* * *
* * * *
* * * * *
```

Code :

```
package module2;

import java.util.Scanner;

public class Q6 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        for(int i=1;i<=n;i++) {
            for(int j=1;j<=n-i;j++) {
                System.out.print(" ");
            }
            for(int k=1;k<=i;k++) {
                System.out.print("* ");
            }
            System.out.println();
        }
        sc.close();
    }
}
```

Output :

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the workspace structure with projects like Aachal, Assignment 2, and module2.
- Code Editor:** Displays the Java file Q6.java containing the following code:

```
1 package module2;
2
3 import java.util.Scanner;
4
5 public class Q6 {
6     public static void main(String[] args) {
7         Scanner sc = new Scanner(System.in);
8         int n = sc.nextInt();
9         for(int i=1;i<=n;i++) {
10             for(int j=1;j<=n-i;j++) {
11                 System.out.print(" ");
12             }
13             *
14             *
15             *
16             *
17             *
18             *
19             *
20             *
21             *
22             *
23             *
24             *
25             *
26             *
27             *
28             *
29             *
30             *
31             *
32             *
33             *
34             *
35             *
36             *
37             *
38             *
39             *
40             *
41             *
42             *
43             *
44             *
45             *
46             *
47             *
48             *
49             *
50             *
51             *
52             *
53             *
54             *
55             *
56             *
57             *
58             *
59             *
60             *
61             *
62             *
63             *
64             *
65             *
66             *
67             *
68             *
69             *
70             *
71             *
72             *
73             *
74             *
75             *
76             *
77             *
78             *
79             *
80             *
81             *
82             *
83             *
84             *
85             *
86             *
87             *
88             *
89             *
90             *
91             *
92             *
93             *
94             *
95             *
96             *
97             *
98             *
99             *
100            *
101            *
102            *
103            *
104            *
105            *
106            *
107            *
108            *
109            *
110            *
111            *
112            *
113            *
114            *
115            *
116            *
117            *
118            *
119            *
120            *
121            *
122            *
123            *
124            *
125            *
126            *
127            *
128            *
129            *
130            *
131            *
132            *
133            *
134            *
135            *
136            *
137            *
138            *
139            *
140            *
141            *
142            *
143            *
144            *
145            *
146            *
147            *
148            *
149            *
150            *
151            *
152            *
153            *
154            *
155            *
156            *
157            *
158            *
159            *
160            *
161            *
162            *
163            *
164            *
165            *
166            *
167            *
168            *
169            *
170            *
171            *
172            *
173            *
174            *
175            *
176            *
177            *
178            *
179            *
180            *
181            *
182            *
183            *
184            *
185            *
186            *
187            *
188            *
189            *
190            *
191            *
192            *
193            *
194            *
195            *
196            *
197            *
198            *
199            *
200            *
201            *
202            *
203            *
204            *
205            *
206            *
207            *
208            *
209            *
210            *
211            *
212            *
213            *
214            *
215            *
216            *
217            *
218            *
219            *
220            *
221            *
222            *
223            *
224            *
225            *
226            *
227            *
228            *
229            *
229            *
230            *
231            *
232            *
233            *
234            *
235            *
236            *
237            *
238            *
239            *
239            *
240            *
241            *
242            *
243            *
244            *
245            *
246            *
247            *
248            *
249            *
249            *
250            *
251            *
252            *
253            *
254            *
255            *
256            *
257            *
258            *
259            *
259            *
260            *
261            *
262            *
263            *
264            *
265            *
266            *
267            *
268            *
269            *
269            *
270            *
271            *
272            *
273            *
274            *
275            *
276            *
277            *
278            *
279            *
279            *
280            *
281            *
282            *
283            *
284            *
285            *
286            *
287            *
288            *
289            *
289            *
290            *
291            *
292            *
293            *
294            *
295            *
296            *
297            *
298            *
299            *
299            *
300            *
301            *
302            *
303            *
304            *
305            *
306            *
307            *
308            *
309            *
309            *
310            *
311            *
312            *
313            *
314            *
315            *
316            *
317            *
318            *
319            *
319            *
320            *
321            *
322            *
323            *
324            *
325            *
326            *
327            *
328            *
329            *
329            *
330            *
331            *
332            *
333            *
334            *
335            *
336            *
337            *
338            *
339            *
339            *
340            *
341            *
342            *
343            *
344            *
345            *
346            *
347            *
348            *
349            *
349            *
350            *
351            *
352            *
353            *
354            *
355            *
356            *
357            *
358            *
359            *
359            *
360            *
361            *
362            *
363            *
364            *
365            *
366            *
367            *
368            *
369            *
369            *
370            *
371            *
372            *
373            *
374            *
375            *
376            *
377            *
378            *
379            *
379            *
380            *
381            *
382            *
383            *
384            *
385            *
386            *
387            *
388            *
389            *
389            *
390            *
391            *
392            *
393            *
394            *
395            *
396            *
397            *
398            *
399            *
399            *
400            *
401            *
402            *
403            *
404            *
405            *
406            *
407            *
408            *
409            *
409            *
410            *
411            *
412            *
413            *
414            *
415            *
416            *
417            *
418            *
419            *
419            *
420            *
421            *
422            *
423            *
424            *
425            *
426            *
427            *
428            *
429            *
429            *
430            *
431            *
432            *
433            *
434            *
435            *
436            *
437            *
438            *
439            *
439            *
440            *
441            *
442            *
443            *
444            *
445            *
446            *
447            *
448            *
449            *
449            *
450            *
451            *
452            *
453            *
454            *
455            *
456            *
457            *
458            *
459            *
459            *
460            *
461            *
462            *
463            *
464            *
465            *
466            *
467            *
468            *
469            *
469            *
470            *
471            *
472            *
473            *
474            *
475            *
476            *
477            *
478            *
479            *
479            *
480            *
481            *
482            *
483            *
484            *
485            *
486            *
487            *
488            *
489            *
489            *
490            *
491            *
492            *
493            *
494            *
495            *
496            *
497            *
498            *
499            *
499            *
500            *
501            *
502            *
503            *
504            *
505            *
506            *
507            *
508            *
509            *
509            *
510            *
511            *
512            *
513            *
514            *
515            *
516            *
517            *
518            *
519            *
519            *
520            *
521            *
522            *
523            *
524            *
525            *
526            *
527            *
528            *
529            *
529            *
530            *
531            *
532            *
533            *
534            *
535            *
536            *
537            *
538            *
539            *
539            *
540            *
541            *
542            *
543            *
544            *
545            *
546            *
547            *
548            *
549            *
549            *
550            *
551            *
552            *
553            *
554            *
555            *
556            *
557            *
558            *
559            *
559            *
560            *
561            *
562            *
563            *
564            *
565            *
566            *
567            *
568            *
569            *
569            *
570            *
571            *
572            *
573            *
574            *
575            *
576            *
577            *
578            *
579            *
579            *
580            *
581            *
582            *
583            *
584            *
585            *
586            *
587            *
588            *
589            *
589            *
590            *
591            *
592            *
593            *
594            *
595            *
596            *
597            *
598            *
599            *
599            *
600            *
601            *
602            *
603            *
604            *
605            *
606            *
607            *
608            *
609            *
609            *
610            *
611            *
612            *
613            *
614            *
615            *
616            *
617            *
618            *
619            *
619            *
620            *
621            *
622            *
623            *
624            *
625            *
626            *
627            *
628            *
629            *
629            *
630            *
631            *
632            *
633            *
634            *
635            *
636            *
637            *
638            *
639            *
639            *
640            *
641            *
642            *
643            *
644            *
645            *
646            *
647            *
648            *
649            *
649            *
650            *
651            *
652            *
653            *
654            *
655            *
656            *
657            *
658            *
659            *
659            *
660            *
661            *
662            *
663            *
664            *
665            *
666            *
667            *
668            *
669            *
669            *
670            *
671            *
672            *
673            *
674            *
675            *
676            *
677            *
678            *
679            *
679            *
680            *
681            *
682            *
683            *
684            *
685            *
686            *
687            *
688            *
689            *
689            *
690            *
691            *
692            *
693            *
694            *
695            *
696            *
697            *
698            *
699            *
699            *
700            *
701            *
702            *
703            *
704            *
705            *
706            *
707            *
708            *
709            *
709            *
710            *
711            *
712            *
713            *
714            *
715            *
716            *
717            *
718            *
719            *
719            *
720            *
721            *
722            *
723            *
724            *
725            *
726            *
727            *
728            *
729            *
729            *
730            *
731            *
732            *
733            *
734            *
735            *
736            *
737            *
738            *
739            *
739            *
740            *
741            *
742            *
743            *
744            *
745            *
746            *
747            *
748            *
749            *
749            *
750            *
751            *
752            *
753            *
754            *
755            *
756            *
757            *
758            *
759            *
759            *
760            *
761            *
762            *
763            *
764            *
765            *
766            *
767            *
768            *
769            *
769            *
770            *
771            *
772            *
773            *
774            *
775            *
776            *
777            *
778            *
779            *
779            *
780            *
781            *
782            *
783            *
784            *
785            *
786            *
787            *
788            *
789            *
789            *
790            *
791            *
792            *
793            *
794            *
795            *
796            *
797            *
798            *
799            *
799            *
800            *
801            *
802            *
803            *
804            *
805            *
806            *
807            *
808            *
809            *
809            *
810            *
811            *
812            *
813            *
814            *
815            *
816            *
817            *
818            *
819            *
819            *
820            *
821            *
822            *
823            *
824            *
825            *
826            *
827            *
828            *
829            *
829            *
830            *
831            *
832            *
833            *
834            *
835            *
836            *
837            *
838            *
839            *
839            *
840            *
841            *
842            *
843            *
844            *
845            *
846            *
847            *
848            *
849            *
849            *
850            *
851            *
852            *
853            *
854            *
855            *
856            *
857            *
858            *
859            *
859            *
860            *
861            *
862            *
863            *
864            *
865            *
866            *
867            *
868            *
869            *
869            *
870            *
871            *
872            *
873            *
874            *
875            *
876            *
877            *
878            *
879            *
879            *
880            *
881            *
882            *
883            *
884            *
885            *
886            *
887            *
888            *
889            *
889            *
890            *
891            *
892            *
893            *
894            *
895            *
896            *
897            *
898            *
898            *
899            *
900            *
901            *
902            *
903            *
904            *
905            *
906            *
907            *
908            *
909            *
909            *
910            *
911            *
912            *
913            *
914            *
915            *
916            *
917            *
918            *
919            *
919            *
920            *
921            *
922            *
923            *
924            *
925            *
926            *
927            *
928            *
929            *
929            *
930            *
931            *
932            *
933            *
934            *
935            *
936            *
937            *
938            *
939            *
939            *
940            *
941            *
942            *
943            *
944            *
945            *
946            *
947            *
948            *
949            *
949            *
950            *
951            *
952            *
953            *
954            *
955            *
956            *
957            *
958            *
959            *
959            *
960            *
961            *
962            *
963            *
964            *
965            *
966            *
967            *
968            *
969            *
969            *
970            *
971            *
972            *
973            *
974            *
975            *
976            *
977            *
978            *
979            *
979            *
980            *
981            *
982            *
983            *
984            *
985            *
986            *
987            *
988            *
989            *
989            *
990            *
991            *
992            *
993            *
994            *
995            *
996            *
997            *
997            *
998            *
999            *
999            *
1000            *
1001            *
1002            *
1003            *
1004            *
1005            *
1006            *
1007            *
1008            *
1009            *
1009            *
1010            *
1011            *
1012            *
1013            *
1014            *
1015            *
1016            *
1017            *
1018            *
1019            *
1019            *
1020            *
1021            *
1022            *
1023            *
1024            *
1025            *
1026            *
1027            *
1028            *
1029            *
1029            *
1030            *
1031            *
1032            *
1033            *
1034            *
1035            *
1036            *
1037            *
1038            *
1039            *
1039            *
1040            *
1041            *
1042            *
1043            *
1044            *
1045            *
1046            *
1047            *
1048            *
1049            *
1049            *
1050            *
1051            *
1052            *
1053            *
1054            *
1055            *
1056            *
1057            *
1058            *
1059            *
1059            *
1060            *
1061            *
1062            *
1063            *
1064            *
1065            *
1066            *
1067            *
1068            *
1069            *
1069            *
1070            *
1071            *
1072            *
1073            *
1074            *
1075            *
1076            *
1077            *
1078            *
1079            *
1079            *
1080            *
1081            *
1082            *
1083            *
1084            *
1085            *
1086            *
1087            *
1088            *
1089            *
1089            *
1090            *
1091            *
1092            *
1093            *
1094            *
1095            *
1096            *
1097            *
1098            *
1098            *
1099            *
1099            *
1100            *
1101            *
1102            *
1103            *
1104            *
1105            *
1106            *
1107            *
1108            *
1109            *
1109            *
1110            *
1111            *
1112            *
1113            *
1114            *
1115            *
1116            *
1117            *
1118            *
1119            *
1119            *
1120            *
1121            *
1122            *
1123            *
1124            *
1125            *
1126            *
1127            *
1128            *
1129            *
1129            *
1130            *
1131            *
1132            *
1133            *
1134            *
1135            *
1136            *
1137            *
1138            *
1139            *
1139            *
1140            *
1141            *
1142            *
1143            *
1144            *
1145            *
1146            *
1147            *
1148            *
1148            *
1149            *
1149            *
1150            *
1151            *
1152            *
1153            *
1154            *
1155            *
1156            *
1157            *
1158            *
1159            *
1159            *
1160            *
1161            *
1162            *
1163            *
1164            *
1165            *
1166            *
1167            *
1168            *
1169            *
1169            *
1170            *
1171            *
1172            *
1173            *
1174            *
1175            *
1176            *
1177            *
1178            *
1179            *
1179            *
1180            *
1181            *
1182            *
1183            *
1184            *
1185            *
1186            *
1187            *
1188            *
1189            *
1189            *
1190            *
1191            *
1192            *
1193            *
1194            *
1195            *
1196            *
1197            *
1198            *
1198            *
1199            *
1199            *
1200            *
1201            *
1202            *
1203            *
1204            *
1205            *
1206            *
1207            *
1208            *
1209            *
1209            *
1210            *
1211            *
1212            *
1213            *
1214            *
1215            *
1216            *
1217            *
1218            *
1219            *
1219            *
1220            *
1221            *
1222            *
1223            *
1224            *
1225            *
1226            *
1227            *
1228            *
1229            *
1229            *
1230            *
1231            *
1232            *
1233            *
1234            *
1235            *
1236            *
1237            *
1238            *
1239            *
1239            *
1240            *
1241            *
1242            *
1243            *
1244            *
1245            *
1246            *
1247            *
1248            *
1249            *
1249            *
1250            *
1251            *
1252            *
1253            *
1254            *
1255            *
1256            *
1257            *
1258            *
1259            *
1259            *
1260            *
1261            *
1262            *
1263            *
1264            *
1265            *
1266            *
1267            *
1268            *
1269            *
1269            *
1270            *
1271            *
1272            *
1273            *
1274            *
1275            *
1276            *
1277            *
1278            *
1279            *
1279            *
1280            *
1281            *
1282            *
1283            *
1284            *
1285            *
1286            *
1287            *
1288            *
1289            *
1289            *
1290            *
1291            *
1292            *
1293            *
1294            *
1295            *
1296            *
1297            *
1298            *
1298            *
1299            *
1299            *
1300            *
1301            *
1302            *
1303            *
1304            *
1305            *
1306            *
1307            *
1308            *
1309            *
1309            *
1310            *
1311            *
1312            *
1313            *
1314            *
1315            *
1316            *
1317            *
1318            *
1319            *
1319            *
1320            *
1321            *
1322            *
1323            *
1324            *
1325            *
1326            *
1327            *
1328            *
1329            *
1329            *
1330            *
1331            *
1332            *
1333            *
1334            *
1335            *
1336            *
1337            *
1338            *
1339            *
1339            *
1340            *
1341            *
1342            *
1343            *
1344            *
1345            *
1346            *
1347            *
1348            *
1349            *
1349            *
1350            *
1351            *
1352            *
1353            *
1354            *
1355            *
1356            *
1357            *
1358            *
1359            *
1359            *
1360            *
1361            *
1362            *
1363            *
1364            *
1365            *
1366            *
1367            *
1368            *
1369            *
1369            *
1370            *
1371            *
1372            *
1373            *
1374            *
1375            *
1376            *
1377            *
1378            *
1379            *
1379            *
1380            *
1381            *
1382            *
1383            *
1384            *
1385            *
1386            *
1387            *
1388            *
1389            *
1389            *
1390            *
1391            *
1392            *
1393            *
1394            *
1395            *
1396            *
1397            *
1398            *
1398            *
1399            *
1399            *
1400            *
1401            *
1402            *
1403            *
1404            *
1405            *
1406            *
1407            *
1408            *
1409            *
1409            *
1410            *
1411            *
1412            *
1413            *
1414            *
1415            *
1416            *
1417            *
1418            *
1419            *
1419            *
1420            *
1421            *
1422            *
1423            *
1424            *
1425            *
1426            *
1427            *
1428            *
1429            *
1429            *
1430            *
1431            *
1432            *
1433            *
1434            *
1435            *
1436            *
1437            *
1438            *
1439            *
1439            *
1440            *
1441            *
1442            *
1443            *
1444            *
1445            *
1446            *
1447            *
1448            *
1449            *
1449            *
1450            *
1451            *
1452            *
1453            *
1454            *
1455            *
1456            *
1457            *
1458            *
1459            *
1459            *
1460            *
1461            *
1462            *
1463            *
1464            *
1465            *
1466            *
1467            *
1468            *
1469            *
1469            *
1470            *
1471            *
1472            *
1473            *
1474            *
1475            *
1476            *
1477            *
1478            *
1479            *
1479            *
1480            *
1481            *
1482            *
1483            *
1484            *
1485            *
1486            *
1487            *
1488            *
1489            *
1489            *
1490            *
1491            *
1492            *
1493            *
1494            *
1495            *
1496            *
1497            *
1498            *
1498            *
1499            *
1499            *
1500            *
1501            *
1502            *
1503            *
1504            *
1505            *
1506            *
1507            *
1508            *
1509            *
1509            *
1510            *
1511            *
1512            *
1513            *
1514            *
1515            *
1516            *
1517            *
1518            *
1519            *
1519            *
1520            *
1521            *
1522            *
1523            *
1524            *
1525            *
1526            *
1527            *
1528            *
1529            *
1529            *
1530            *
1531            *
1532            *
1533            *
1534            *
1535            *
1536            *
1537            *
1538            *
1539            *
1539            *
1540            *
1541            *
1542            *
1543            *
1544            *
1545            *
1546            *
1547            *
1548            *
1549            *
1549            *
1550            *
1551            *
1552            *
1553            *
1554            *
1555            *
1556            *
1557            *
1558            *
1559            *
1559            *
1560            *
1561            *
1562            *
1563            *
1564            *
1565            *
1566            *
1567            *
1568            *
1569            *
1569            *
1570            *
1571            *
1572            *
1573            *
1574            *
1575            *
1576            *
1577            *
1578            *
1579            *
1579            *
1580            *
1581            *
1582            *
1583            *
1584            *
1585            *
1586            *
1587            *
1588            *
1589            *
1589            *
1590            *
1591            *
1592            *
1593            *
1594            *
1595            *
1596            *
1597            *
1598            *
1598            *
1599            *
1599            *
1600            *
1601            *
1602            *
1603            *
1604            *
1605            *
1606            *
1607            *
1608            *
1609            *
1609            *
1610            *
1611            *
1612            *
1613            *
1614            *
1615            *
1616            *
1617            *
1618            *
1619            *
1619            *
1620            *
1621            *
1622            *
1623            *
1624            *
1625            *
1626            *
1627            *
1628            *
1629            *
1629            *
1630            *
1631            *
1632            *
1633            *
1634            *
1635            *
1636            *
1637            *
1638            *
1639            *
1639            *
1640            *
1641            *
1642            *
1643            *
1644            *
1645            *
1646            *
1647            *
1648            *
1649            *
1649            *
1650            *
1651            *
1652            *
1653            *
1654            *
1655            *
1656            *
1657            *
1658            *
1659            *
1659            *
1660            *
1661            *
1662            *
1663            *
1664            *
1665            *
1666            *
1667            *
1668            *
1669            *
1669            *
1670            *
1671            *
1672            *
1673            *
1674            *
1675            *
1676            *
1677            *
1678            *
1679            *
1679            *
1680            *
1681            *
1682            *
1683            *
1684            *
1685            *
1686            *
1687            *
1688            *
1689            *
1689            *
1690            *
1691            *
1692            *
1693            *
1694            *
1695            *
1696            *
1697            *
1698            *
1698            *
1699            *
1699            *
1700            *
1701            *
1702            *
1703            *
1704            *
1705            *
1706            *
1707            *
1708            *
1709            *
1709            *
1710            *
1711            *
1712            *
1713            *
1714            *
1715            *
1716            *
1717            *
1718            *
1719            *
1719            *
1720            *
1721            *
1722            *
1723            *
1724            *
1725            *
1726            *
1727            *
1728            *
1729            *
1729            *
1730            *
1731            *
1732            *
1733            *
1734            *
1735            *
1736            *
1737            *
1738            *
1739            *
1739            *
1740            *
1741            *
1742            *
1743            *
1744            *
1745            *
1746            *
1747            *
1748            *
1749            *
1749            *
1750            *
1751            *
1752            *
1753            *
1754            *
1755            *
1756            *
1757            *
1758            *
1759            *
1759            *
1760            *
1761            *
1762            *
1763            *
1764            *
1765            *
1766            *
1767            *
1768            *
1769            *
1769            *
1770            *
1771            *
1772            *
1773            *
1774            *
1775            *
1776            *
1777            *
1778            *
1779            *
1779            *
1780            *
1781            *
1782            *
1783            *
1784            *
1785            *
1786            *
1787            *
1788            *
1789            *
1789            *
1790            *
1791            *
1792            *
1793            *
1794            *
1795            *
1796            *
1797            *
1798            *
1798            *
1799            *
1799            *
1800            *
1801            *
1802            *
1803            *
1804            *
1805            *
1806            *
1807            *
1808            *
1809            *
1809            *
1810            *
1811            *
1812            *
1813            *
1814            *
1815            *
1816            *
1817            *
1818            *
1819            *
1819            *
1820            *
1821            *
1822            *
1823            *
1824            *
1825            *
1826            *
1827            *
1828            *
1829            *
1829            *
1830            *
1831            *
1832            *
1833            *
1834            *
1835            *
1836            *
1837            *
1838            *
1839            *
1839            *
1840            *
1841            *
1842            *
1843            *
1844            *
1845            *
1846            *
1847            *
1848            *
1849            *
1849            *
1850            *
1851            *
1852            *
1853            *
1854            *
1855            *
1856            *
1857            *
1858            *
1859            *
1859            *
1860            *
1861            *
1862            *
1863            *
1864            *
1865            *
1866            *
1867            *
1868            *
1869            *
1869            *
1870            *
1871            *
1872            *
18
```

Q7. Write a Java program that will read in month and day (as numerical value). The program will return the equivalent zodiac sign.

Zodiac Sign	Duration
Aquarius	Jan 20 – Feb 18
Pisces	Feb 19 – Mar 20
Aries	Mar 21 – Apr 19
Taurus	Apr 20 – May 20
Gemini	May 21 – Jun 20
Cancer	Jun 21 – Jul 22
Leo	Jul 23 – Aug 22
Virgo	Aug 23 – Sep 22
Libra	Sep 23 – Oct 22
Scorpio	Oct 23 – Nov 21
Sagittarius	Nov 22 – Dec 21
Capricorn	Dec 22 – Jan 19

Code :

```

package module2;

import java.util.Scanner;

public class Q7 {
    public static void main(String[] args) {
        enum Zodiac{Aquarius, Pisces, Aries, Taurus,
                   Gemini,Cancer,Leo,Virgo,Libra,Scorpio,Sagittarius,
                   Capricorn};
        enum Month{January,February,March,April,May,June,July,August,
                   September,October,November,December};
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the Day : ");
        int day = sc.nextInt();
        System.out.println("Enter the Month (1-12) : ");
        int Mon = sc.nextInt();
        Month M[] = Month.values();
        Zodiac z[] = Zodiac.values();
        Mon=Character.toLowerCase(Mon);
        if(Mon==1)
        {
            if(day>=1 && day<=19)
                System.out.println("Zodiac sign for "+M[Mon-1]+" "+day+" is
"+z[11]);
            if(day>=20 && day<=31)
                System.out.println("Zodiac sign for "+M[Mon-1]+" "+day+" is
"+z[Mon]);
        }
    }
}

```

```
else if(Mon==2)
{
    if(day>=1 && day<=18)
        System.out.println("Zodiac sign for "+M[Mon-1]+" "+day+
is "+z[Mon-1]);
    if(day>=19 && day<=29)
        System.out.println("Zodiac sign for "+M[Mon-1]+" "+day+
is "+z[Mon]);
}else if(Mon==3)
{
    if(day>=1 && day<=20)
        System.out.println("Zodiac sign for "+M[Mon-1]+" "+day+
is "+z[Mon-1]);
    if(day>=21 && day<=31)
        System.out.println("Zodiac sign for "+M[Mon-1]+" "+day+
is "+z[Mon]);
}else if(Mon==4)
{
    if(day>=1 && day<=19)
        System.out.println("Zodiac sign for "+M[Mon-1]+" "+day+
is "+z[Mon-1]);
    if(day>=20 && day<=30)
        System.out.println("Zodiac sign for "+M[Mon-1]+" "+day+
is "+z[Mon]);
}else if(Mon==5)
{
    if(day>=1 && day<=20)
        System.out.println("Zodiac sign for "+M[Mon-1]+" "+day+
is "+z[Mon-1]);
    if(day>=21 && day<=31)
        System.out.println("Zodiac sign for "+M[Mon-1]+" "+day+
is "+z[Mon]);
}else if(Mon==6)
{
    if(day>=1 && day<=20)
        System.out.println("Zodiac sign for "+M[Mon-1]+" "+day+
is "+z[Mon-1]);
    if(day>=21 && day<=30)
        System.out.println("Zodiac sign for "+M[Mon-1]+" "+day+
is "+z[Mon]);
}else if(Mon==7)
{
    if(day>=1 && day<=22)
        System.out.println("Zodiac sign for "+M[Mon-1]+" "+day+
is "+z[Mon-1]);
    if(day>=23 && day<=31)
        System.out.println("Zodiac sign for "+M[Mon-1]+" "+day+
is "+z[Mon]);
}
```

```
else if(Mon==8)
{
    if(day>=1 && day<=22)
        System.out.println("Zodiac sign for "+M[Mon-1]+
"+day+ is "+z[Mon-1]);
    if(day>=23 && day<=31)
        System.out.println("Zodic sign for "+M[Mon-1]+
"+day+ is "+z[Mon]);
}

else if(Mon==9)
{
    if(day>=1 && day<=22)
        System.out.println("Zodiac sign for "+M[Mon-1]+
"+day+ is "+z[Mon-1]);
    if(day>=23 && day<=30)
        System.out.println("Zodic sign for "+M[Mon-1]+
"+day+ is "+z[Mon]);
}else if(Mon==10)
{
    if(day>=1 && day<=22)
        System.out.println("Zodiac sign for "+M[Mon-1]+
"+day+ is "+z[Mon-1]);
    if(day>=23 && day<=31)
        System.out.println("Zodic sign for "+M[Mon-1]+
"+day+ is "+z[Mon]);
}else if(Mon==11)
{
    if(day>=1 && day<=21)
        System.out.println("Zodiac sign for "+M[Mon-1]+
"+day+ is "+z[Mon-1]);
    if(day>=22 && day<=30)
        System.out.println("Zodic sign for "+M[Mon-1]+
"+day+ is "+z[Mon]);
}else if(Mon==12)
{
    if(day>=1 && day<=21)
        System.out.println("Zodiac sign for "+M[Mon-1]+
"+day+ is "+z[Mon-1]);
    if(day>=22 && day<=31)
        System.out.println("Zodic sign for "+M[Mon-1]+
"+day+ is "+z[Mon]);
}
sc.close();
}
```

Output :

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure with packages like Assignment 2, module2, and src, containing various Java files (Q1.java through Q16.java, Q2.java, Q3.java, Q4.java, Q5.java, Q6.java, Q7.java, Q8.java, Q9.java, and Student.java).
- Code Editor:** Displays the content of Q7.java, which defines a class Q7 with a main method. The main method uses enums for Zodiac signs and months to determine the zodiac sign for a given date.
- Console Output:** Shows the terminal output of the application's execution. It prompts for the day (Enter the Day :), receives input 5, prompts for the month (Enter the Month (1-12) :), receives input 6, and then outputs "Zodiac sign for June 5 is Cancer".
- Bottom Status Bar:** Provides system information including temperature (40°C Haze), time (04:52), and date (10-04-2022).

Q8. Write Java code statements that accomplish the tasks listed below

- a) Declare an array of integers.
- b) Allocate storage to allow 5 integers to be stored in the array.
- c) Populate the array with the values: 1, 8, 27, 64, 125
- d) Replace the third array element with the value -7.
- e) Copy the value of the fifth array element to the first array storage location.
- f) Subtract the value stored in the second array storage location from the value stored in the
- g) third and store the difference in the fourth array storage location.
- h) Compute the sum of the array elements with subscripts 1 to 3.

Code :

```
package module2;

public class Q8 {
    public static void main(String[] args) {
        int b=0,sum=0;
        System.out.println("Task a");
        int a[];
        System.out.println("Array of integer is declared.\n");
        System.out.println("Task b");
        a = new int[5];
        System.out.println("Storage of 5 is allocated.\n");
        System.out.println("Task c");
        for(int i=0;i<5;i++)
        {
            b=i+1;
            a[i]=(b*b*b);
        }
        System.out.println("Data stored in array.\n");
        System.out.println("Task d");
        a[2]=-7;
        System.out.println("3rd element is replaced to -7.\n");
        System.out.println("Task e");
        a[0] = a[4];
        System.out.println("value of 5th element is copied in
1st.\n");
        System.out.println("Task f");
        a[3] = a[2] - a[1];
        System.out.print("second element value is subtracted from
value of");
        System.out.println(" thrid element and stored in fouth
element.\n");
        System.out.println("Task g");
        for(int i = 1 ; i <= 3 ; i++) {
            sum+=a[i];
        }
        System.out.println("Sum of subscript 1 to 3 = "+sum);
        System.out.println("-----");
        System.out.println("All tasks completed");
    }
}
```

Output :

```
eclipse workspace - Assignment 2/src/module2/Q8.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
File Edit Source Refactor Navigate Search Project Run Window Help
Markers Properties Servers Data Source Explorer Snippets Console Coverage
terminated> Q8 [Java Application] C:\Users\Devi\p2\pool\plugins\org.eclipse.jdt.core\openjdkhotspot\refull.win32\x86_64_16.0.2.v20210721-1149\jre\bin\javaw.exe (10-Apr-2022, 9:17:31 am - 9:17:32 am)
Task a
Array of integer is declared.

Task b
Storage of 5 is allocated.

Task c
Data stored in array.

Task d
3rd element is replaced to -7.

Task e
value of 5th element is copied in 1st.

Task f
second element value is subtracted from value of thrid element and stored in fouth element.

Task g
Sum of subscript 1 to 3 = -14
-----
All tasks completed
```

Q9. Design a Java code to receive input for 'n' numbers and sort numbers based on user's choice either ascending or descending. Finally print results.

Code :

```
package module2;

import java.util.Scanner;

public class Q9 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int[] a = {1,2,3,9,8,7,6,4,53,4};
        int temp=0;
        System.out.println("Array before Sorting...");
        for(int i : a)
            System.out.print(i+" ");
        System.out.println();
        System.out.println("-----");
        //sorting array Arrays.sort(a);
        for(int i = 0 ; i < a.length ; i++) {
            for(int j = i+1 ; j < a.length ; j++) {
                if(a[i] > a[j]) {
                    temp = a[i];
                    a[i] = a[j];
                    a[j] = temp;
                }
            }
        }
        System.out.println("For Sorting array in Ascending order press a");
        System.out.println("For Sorting array in Descending order press d");
        System.out.print("=> ");
        char c = sc.next().charAt(0);
        System.out.println("-----");
        if('a' == Character.toLowerCase(c)) {
            System.out.println("Ascending order...");
            for(int b : a)
                System.out.print(b+" ");
        }
        else if('d' == Character.toLowerCase(c)) {
            System.out.println("Descending order...");
            for(int i = a.length - 1; i >= 0; i--)
                System.out.print(a[i]+" ");
        }
        sc.close();
    }
}
```

Output :

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the workspace structure with projects like Aachal, Assignment 2, and various Java files under module2.
- Editor:** Displays the Java code for Q9.java:

```
1 package module2;
2
3 import java.util.Scanner;
4
5 public class Q9 {
6     public static void main(String[] args) {
7         Scanner sc = new Scanner(System.in);
8         int[] a = {1,2,3,9,8,7,6,4,53,4};
9         int temp=0;
10        System.out.println("Array before Sorting...");
11        for(int i : a)
```
- Console:** Shows the output of the program:

```
Array before Sorting...
1 2 3 9 8 7 6 4 53 4
-----
For Sorting array in Ascending order press a
For Sorting array in Descending order press d
=> a
|-----
```
- Bottom Status Bar:** Shows system information including temperature (40°C), battery level (Haze), and date/time (10-Apr-2022, 9:11:34 am).

Q10. Design a Java program to get a string and do the following in the same program.

- 1) Get a character. Find the occurrence of the character from right and left side. Display that information separately.
- 2) Get a positive integer from user and find the character of the index such that should not create run time error.

Code :

```
package module2;
import java.util.Scanner;
public class Q10 {
    public static void main(String[] args) {
        boolean foundChar = false;
        boolean intFound = false;
        System.out.print("Enter the string : ");
        Scanner sc = new Scanner(System.in);
        String str = sc.next();
        System.out.print("Enter the Character :");
        char ch = sc.next().charAt(0);
        System.out.println("Searching input character in string...");
        for(int i = 0 ; i < str.length() ; i++) {
            if(ch == str.charAt(i)) {
                System.out.print(ch+" is found at location : "+(str.length()-i));
                System.out.println(" from right and " +(i+1)+" from left.");
                foundChar = true;
            }
        }
        if(foundChar == false) {
            System.out.println("Given character is not found ");
            System.out.print("Program is terminating...");
            System.exit(0);
        }
        System.out.println("Task 1 Completed");
        System.out.println("-----\n");

        System.out.print("Enter positive number between 0 to "+(str.length()-1));
        int a = sc.nextInt();
        if(a < str.length()) {
            intFound = true;
        }
        if(intFound == false) {
            System.out.println("Given index is out of range");
            System.out.print("Program is terminating...");
            System.exit(0);
        }
        System.out.println("Searching for character for given index...");
        System.out.println("Character at index "+a+" is : "+str.charAt(a));
        System.out.println("Task 2 Completed");
        sc.close();
    }
}
```

Output :

```
eclipse workspace - Assignment 2/src/module2/Q10.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Markers Properties Servers Data Source Explorer Snippets Console Coverage
terminated> Q10 Java Application C:\Users\DELL.p2pool\plugins\org.eclipse.jst\openjdk.hotspot.jre.full.win32.x86_64_16.0.2.v20210721-1149\jre\bin\javaw.exe (10-Apr-2022, 9:25:18 am - 9:25:38 am)
# Enter the string : sarang
Enter the Character :a
Searching input character in string...
a is found at location : 5 from right and 2 from left.
a is found at location : 3 from right and 4 from left.
Task 1 Completed
-----
Enter positive number between 0 to 53
Searching for character for given index...
Character at index 3 is : a
Task 2 Completed
```

Q11. Design a Java program to get ‘n’ numbers and a number. Apply the linear search and binary search. Find the best algorithm through the computation and display the result.

Code :

```
package module2;

import java.util.Arrays;
import java.util.Scanner;

public class Q11 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int a[] = new int[5];
        int temp = 0;
        //input
        System.out.println("Enter 5 random numbers : ");
        for(int i = 0 ; i < a.length ; i++)
            a[i] = sc.nextInt();
        //sorting
        for(int i = 0 ; i < a.length ; i++) {
            for(int j = i+1 ; j < a.length ; j++) {
                if(a[i] > a[j]) {
                    temp = a[i];
                    a[i] = a[j];
                    a[j] = temp;
                }
            }
        }
        //Key
        int key = sc.nextInt();
        //Linear Search
        long start, end;
        start = System.nanoTime();
        for(int i = 0 ; i < a.length ; i++) {
            if(a[i]==key) {
                break;
            }
        }
        end = System.nanoTime();
        long linearEfficiency = end - start;

        //Binary Search
        start = System.nanoTime();
        Arrays.binarySearch(a, key);
        end = System.nanoTime();
        long binaryEfficiency = end - start;
```

```
//output
System.out.println("Result : ");
System.out.println("-----");
System.out.println("Computation efficiency of linear search : " +
linearEfficiency+"ns and Binary search : "+binaryEfficiency+"ns");
System.out.println("-----");
System.out.println("Conclusion :");
System.out.println("-----");
if(linearEfficiency < binaryEfficiency)
    System.out.println("linear search is reliable than binary search");
else
    System.out.println("binary search is reliable than linear search");
sc.close();

}
}
```

Output :

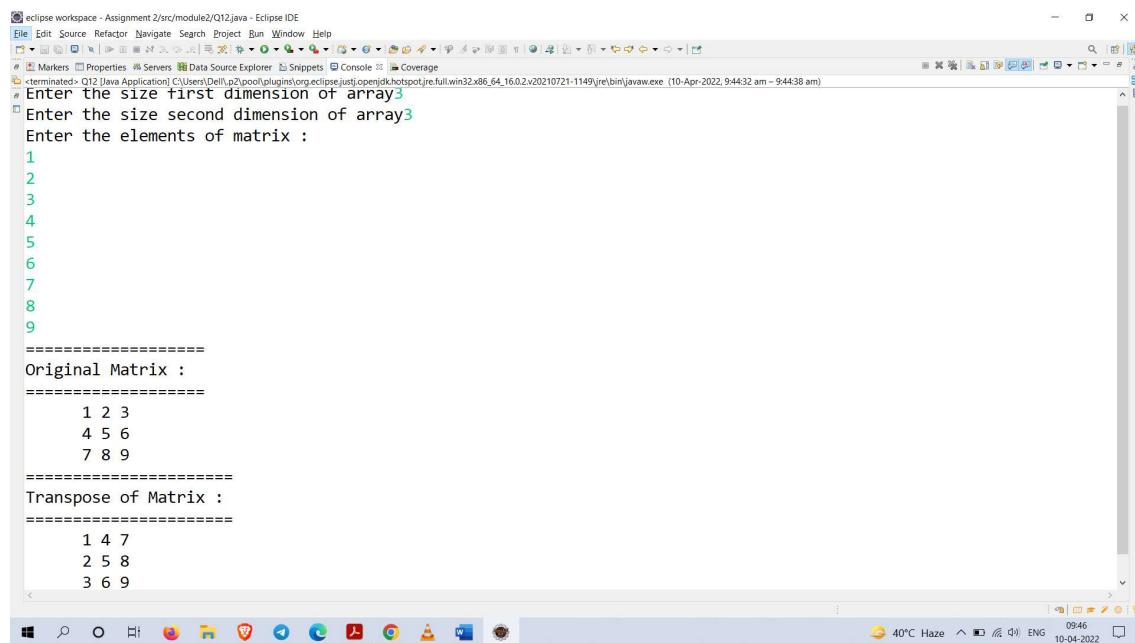
```
eclipse workspace - Assignment 2/src/module2/Q11.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Markers Properties Servers Data Source Explorer Snippets Console Coverage
<terminated> Q11 [Java Application] C:\Users\Deell.p2\pool\plugins\org.eclipse.jst\openjdk.hotspot.jre.full.win32\x86_64_16.0.2.v20210721-1149\jre\bin\javaw.exe (10-Apr-2022, 9:30:53 am - 9:31:06 am)
Enter 5 random numbers :
10
52
36
12
45
31
Result :
-----
Computation efficiency of linear search : 1500ns and Binary search : 872100ns
-----
Conclusion :
-----
linear search is reliable than binary search
```

Q12. Design a Java Program to get a matrix as input and print the transpose of the given matrix.

Code :

```
package module2;
import java.util.Scanner;
public class Q12 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the size first dimension of array");
        int n1 = sc.nextInt();
        System.out.print("Enter the size second dimension of array");
        int n2 = sc.nextInt();
        int a[][] = new int[n1][n2];
        int b[][] = new int[n1][n2];
        System.out.println("Enter the elements of matrix : ");
        //input elements of matrix
        for(int i = 0 ; i < n1 ; i++) {
            for(int j = 0 ; j < n2 ; j++) {
                a[i][j] = sc.nextInt();
            }
        }
        //display original matrix
        System.out.println("=====");
        System.out.println("Original Matrix : ");
        System.out.println("=====");
        for(int x = 0 ; x < a.length; x++) {
            System.out.print("      ");
            for(int y = 0 ; y < a.length; y++)
                System.out.print(a[x][y]+" ");
            System.out.println();
        }
        //Transpose
        for(int i = 0 ; i < n1 ; i++) {
            for(int j = 0 ; j < n2 ; j++) {
                b[i][j] = a[j][i];
            }
        }
        //display transpose output
        System.out.println("=====");
        System.out.println("Transpose of Matrix : ");
        System.out.println("=====");
        for(int i = 0 ; i < n1 ; i++) {
            System.out.print("      ");
            for(int j = 0 ; j < n2 ; j++) {
                System.out.print(b[i][j]+" ");
            }
            System.out.println();
        }
        sc.close();
    }
}
```

Output :



eclipse workspace - Assignment 2/src/module2/Q12.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Markers Properties Servers Data Source Explorer Snippets Console Coverage

terminated> Q12 [Java Application] C:\Users\Deepti.p2pool\plugins\org.eclipse.jdt.core\openjdk\hotspot\jre\full\win32\x86_64_16.0.2.v20210721-1149\jre\bin\javaw.exe (10-Apr-2022, 9:44:32 am - 9:44:38 am)

» Enter the size first dimension of array3

» Enter the size second dimension of array3

Enter the elements of matrix :

```
1
2
3
4
5
6
7
8
9
=====
Original Matrix :
=====
1 2 3
4 5 6
7 8 9
=====
Transpose of Matrix :
=====
1 4 7
2 5 8
3 6 9
```

Windows taskbar at the bottom:

- Start button
- Search icon
- Task view icon
- Firefox icon
- File Explorer icon
- OneDrive icon
- Taskbar separator
- Calculator icon
- Taskbar separator
- Google Chrome icon
- Taskbar separator
- VLC icon
- Taskbar separator
- Windows Update icon
- Taskbar separator
- System tray icons: battery (40°C Haze), network, volume, clock (09:46), date (10-04-2022)

Q13. Develop a Java program to get a matrix and print the lower triangle of the matrix. Apply the necessary conditions if it required.

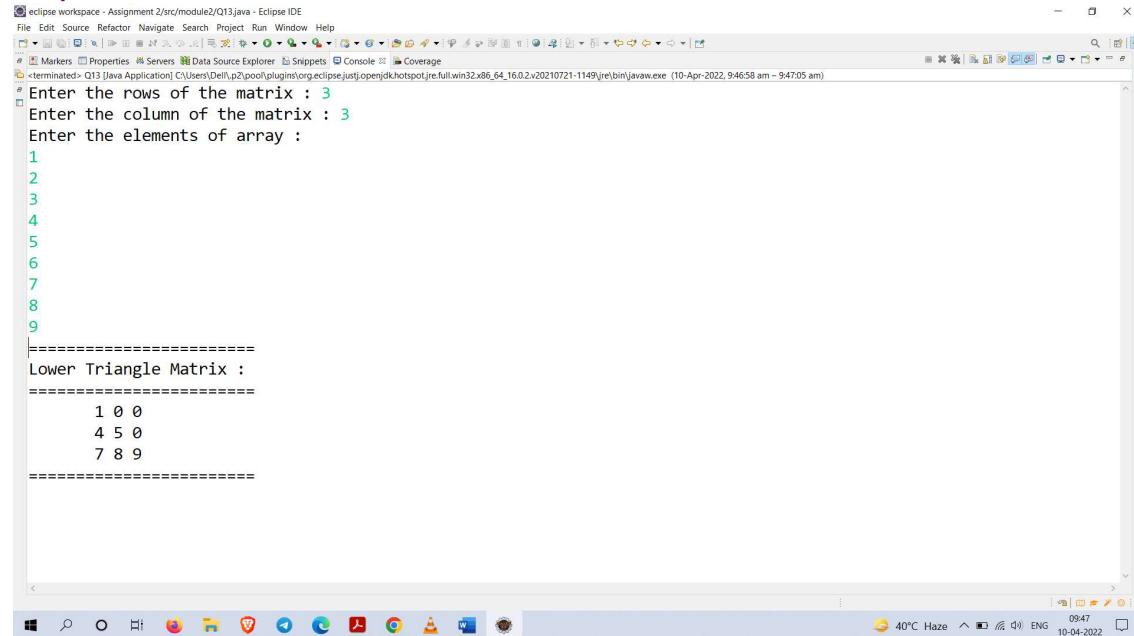
Code :

```
package module2;

import java.util.Scanner;

public class Q13 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n1,n2;
        //no of rows
        System.out.print("Enter the rows of the matrix : ");
        n1 = sc.nextInt();
        //no of columns
        System.out.print("Enter the column of the matrix : ");
        n2 = sc.nextInt();
        //creating array of dimensions n1 x n2
        int a[][] = new int[n1][n2];
        //taking input
        System.out.println("Enter the elements of array : ");
        for(int i = 0 ; i < n1 ; i++) {
            for(int j = 0 ; j < n2 ; j++) {
                a[i][j] = sc.nextInt();
            }
        }
        //printing the matrix
        System.out.println("=====");
        System.out.println("Lower Triangle Matrix : ");
        System.out.println("=====");
        for(int i = 0 ; i < n1 ; i++) {
            System.out.print("    ");
            for(int j = 0 ; j < n2 ; j++) {
                if(i < j) {
                    a[i][j] = 0;
                }
                System.out.print(a[i][j]+" ");
            }
            System.out.println();
        }
        System.out.println("=====");
        sc.close();
    }
}
```

Output :



The screenshot shows the Eclipse IDE interface with a Java application running. The console tab displays the following output:

```
# eclipse workspace - Assignment 2/src/module2/Q13.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Markers Properties Servers Data Source Explorer Snippets Console Coverage
<terminated> O13 [Java Application] C:\Users\Delli.p2\pool\plugins\org.eclipse.jst\openjdk.hotspot.jre.full.win32.x86_64_16.0.2.v20210721-1149\jre\bin\javaw.exe (10-Apr-2022, 9:46:58 am - 9:47:05 am)
# Enter the rows of the matrix : 3
Enter the column of the matrix : 3
Enter the elements of array :
1
2
3
4
5
6
7
8
9
=====
Lower Triangle Matrix :
=====
1 0 0
4 5 0
7 8 9
=====
```

The Java application prompts for the number of rows and columns, then asks for the elements of the array. It then prints the resulting lower triangle matrix.

Q14. Get 'n' integer numbers from user and find the count of each unique number. Display the result as number – its count. Example: 3 - 5, where 3 is the number presented for 5 times)

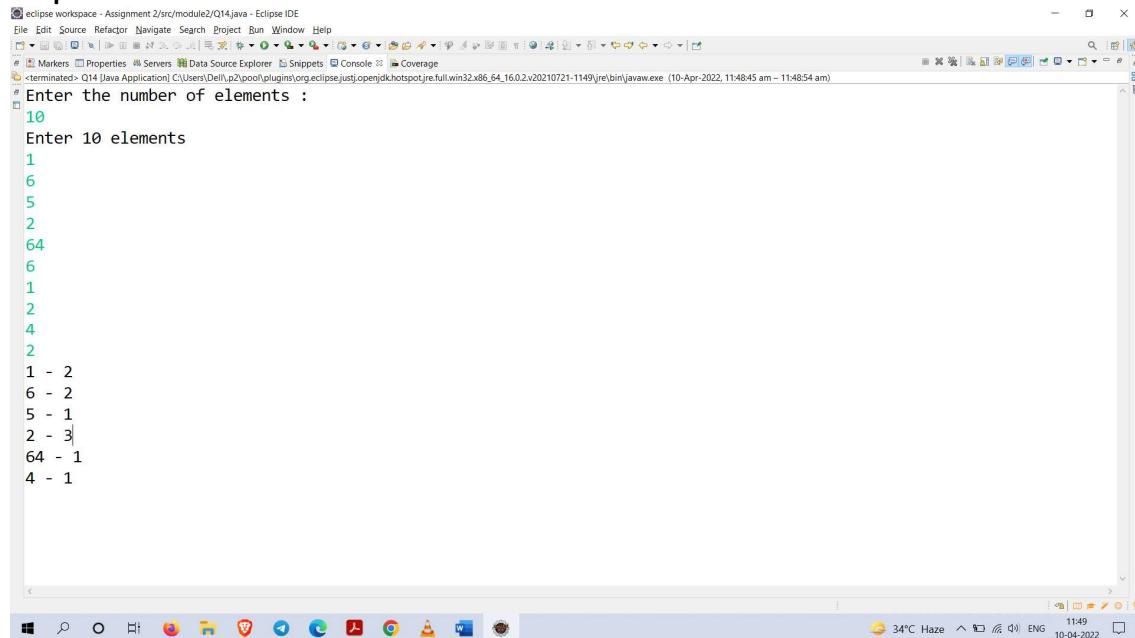
Code :

```
package module2;

import java.util.Scanner;

public class Q14 {
    public static void main(String[] args) {
        int n = 0;
        int count = 0;
        boolean check = false;
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the number of elements : ");
        n = sc.nextInt();
        int a[] = new int[n];
        int i = 0;
        System.out.println("Enter "+n+" elements");
        for(@SuppressWarnings("unused") int x : a) {
            a[i] = sc.nextInt();
            i++;
        }
        for(int k = 0 ; k < a.length ; k++) {
            check =false;
            count = 0;
            for(int j = k ; j < a.length ; j++) {
                for(int z = 0 ; z < k ; z++) {
                    if(a[z] == a[k])
                    {
                        check = true;
                        break;
                    }
                }
                if(check == true)
                    continue;
                if(a[k] == a[j])
                    count++;
            }
            if(check == false)
                System.out.println(a[k]+" - "+count);
        }
        sc.close();
    }
}
```

Output :



eclipse workspace - Assignment 2/src/module2/Q14.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Markers Properties Servers Data Source Explorer Snippets Console Coverage

terminated> Q14 Java Application C:\Users\DELL\p2pool\plugins\org.eclipse.jdt.openjdk.hotspot.re.full.win32.x86_64_16.0.2.v20210721-1149\jre\bin\javaw.exe (10-Apr-2022, 11:48:45 am - 11:48:54 am)

```
# Enter the number of elements :
10
Enter 10 elements
1
6
5
2
64
6
1
2
4
2
1 - 2
6 - 2
5 - 1
2 - 3
64 - 1
4 - 1
```

Windows taskbar at the bottom:

- Icons for File Explorer, Search, Task View, Taskbar settings, Edge, Firefox, File History, Task Manager, Task View, Taskbar settings, Taskbar settings.
- System tray icons: Battery (34°C Haze), Network, Volume, Language (ENG), Date (10-04-2022), Time (11:49).

Q15. Write a Java program to get a string from user. Divide the given string into 5 equals parts and make those part as new string. At last, print all information. Apply the necessary conditions

Code :

```
package module2;

import java.util.Scanner;

public class Q15 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String str = sc.nextLine();
        if(str.length()%5!=0) {
            System.out.println(str.length());
            System.out.println("Cannot be divided into 5
parts.");
            System.exit(0);
        }
        String[] sho = new String[5];
        int j = 0 , count = 0 ;
        sho[j]="";
        for(int i = 0 ; i < str.length() ; i++) {
            if(count == (str.length()/5)) {
                j++;
                count = 0;
                sho[j]="";
            }
            count++;
            sho[j] += str.charAt(i);
        }
        System.out.println("\nOutput : ");
        System.out.println("-----");
        for(String x : sho)
            System.out.println(x);
        sc.close();
    }
}
```

Output :

The screenshot shows the Eclipse IDE interface with the title bar "eclipse workspace - Assignment 2/src/module2/Q15.java - Eclipse IDE". The menu bar includes File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, and Help. The toolbar has various icons for file operations. The left sidebar shows markers, properties, servers, data source explorer, snippets, and coverage. The central workspace shows a terminal window with the command "java Q15" and the output "saran g gir ish d eodha r1234". The bottom status bar shows system information like temperature (34°C), battery level (Haze), and date/time (10-04-2022).

```
saran
g gir
ish d
eodha
r1234
```

Q16. Design a Java Program to represent time as class which contains the following:

- Hours, minutes and seconds as members.
- Basic Methods to get input, print the time.
- Method to find the difference between two times.

Demonstrate these methods through few objects. Also apply the necessary conditions

Code :

```
package module2;
import java.util.Scanner;
public class Q16 {
    static Scanner sc;
    private int hours;
    private int minutes;
    private int seconds;
    private int valid = 1;
    private static int count = 0;
    static Q16 end;
    static Q16 start;
    static Q16 diff;
    public void input() {
        sc = new Scanner(System.in);
        while(valid==1) {
            valid = 0;
            if(count == 0) {
                System.out.println("Starting Time : ");
            }
            else if(count == 1) {
                System.out.println("Ending Time : ");
            }
            System.out.print("Enter the hours, minutes and seconds : ");
            hours = sc.nextInt();
            minutes = sc.nextInt();
            seconds = sc.nextInt();
            if(hours > 24 || hours < 0 || minutes > 60 || minutes < 0 ||
            seconds > 60 || seconds < 0)
            {
                System.out.println("\nInvalid input\n");
                valid = 1;
            }
            if(count==1) {
                if(end.hours < start.hours) {
                    System.out.println("\nInvalid input");
                    System.out.println("End time must be greater than
start time\n");
                    valid = 1;
                }
            }
        }
    }
    public void print() {
        System.out.println(hours+" "+minutes+" "+seconds);
    }
}
```

```
public void difference() {
    this.hours = end.hours - start.hours;
    this.minutes = end.minutes - start.minutes;
    this.seconds = end.seconds - start.seconds;
    if(seconds < 0) {
        seconds += 60;
        minutes -= 1;
    }
    if(minutes < 0) {
        minutes += 60;
        hours -= 1;
    }
    System.out.println("time difference : ");
    System.out.println(hours+":"+minutes+":"+seconds);
    System.out.println("\n");
}
public static void main(String[] args) {
    for (int i = 0; i<3; i++)
    {
        System.out.println("For iteration - "+i+" : ");
        System.out.println("=====");
        start = new Q16();
        end = new Q16();
        diff = new Q16();
        start.input();
        count++;
        end.input();
        System.out.println("Starting Time : ");
        start.print();
        System.out.println("Ending Time : ");
        end.print();
        diff.difference();
        count=0;
    }
    sc.close();
}
```

Output :

Part 1

```
eclipse workspace - Assignment 2/src/module2/Q16.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Markers Properties Servers Data Source Explorer Snippets Console Coverage
terminated> Q16 [Java Application] C:\Users\Deell\p2\pool\plugins\org.eclipse.jdt.openjdk.hotspot.jre.full.win32.x86_64_16.0.2.v20210721-1149\jre\bin\javaw.exe (10-Apr-2022, 12:04:52 pm - 12:05:32 pm)

# For iteration - 0 :
=====
Starting Time :
Enter the hours, minutes and seconds : 12 0
0
Ending Time :
Enter the hours, minutes and seconds : 10 20 20

Invalid input
End time must be greater than start time

Ending Time :
Enter the hours, minutes and seconds : 13 20 20
Starting Time :
12 0 0
Ending Time :
13 20 20
time difference :
1:20:20

For iteration - 1 :
=====
Starting Time :
```

Part 2

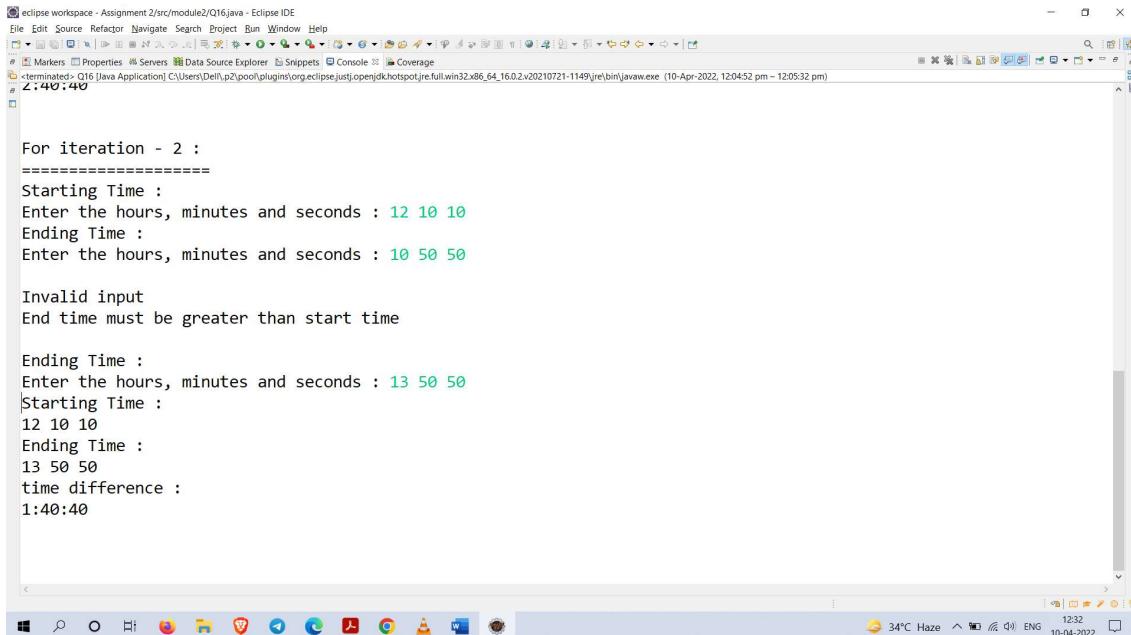
```
eclipse workspace - Assignment 2/src/module2/Q16.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Markers Properties Servers Data Source Explorer Snippets Console Coverage
terminated> Q16 [Java Application] C:\Users\Deell\p2\pool\plugins\org.eclipse.jdt.openjdk.hotspot.jre.full.win32.x86_64_16.0.2.v20210721-1149\jre\bin\javaw.exe (10-Apr-2022, 12:04:52 pm - 12:05:32 pm)

13 20 20
time difference :
1:20:20

For iteration - 1 :
=====
Starting Time :
Enter the hours, minutes and seconds : 10 10 10
Ending Time :
Enter the hours, minutes and seconds : 12 50 50
Starting Time :
10 10 10
Ending Time :
12 50 50
time difference :
2:40:40

For iteration - 2 :
=====
Starting Time :
Enter the hours, minutes and seconds : 12 10 10
Ending Time :
```

Part 3



The screenshot shows the Eclipse IDE interface with a Java application running in the Console view. The console output is as follows:

```
For iteration - 2 :  
=====  
Starting Time :  
Enter the hours, minutes and seconds : 12 10 10  
Ending Time :  
Enter the hours, minutes and seconds : 10 50 50  
  
Invalid input  
End time must be greater than start time  
  
Ending Time :  
Enter the hours, minutes and seconds : 13 50 50  
Starting Time :  
12 10 10  
Ending Time :  
13 50 50  
time difference :  
1:40:40
```

The status bar at the bottom right of the window shows the date as 10-04-2022, time as 12:32, and weather as 34°C Haze.

Q18. Design a Java program with your choice of a class to represent any real-world object. Illustrate *constructors* and *this* keyword.

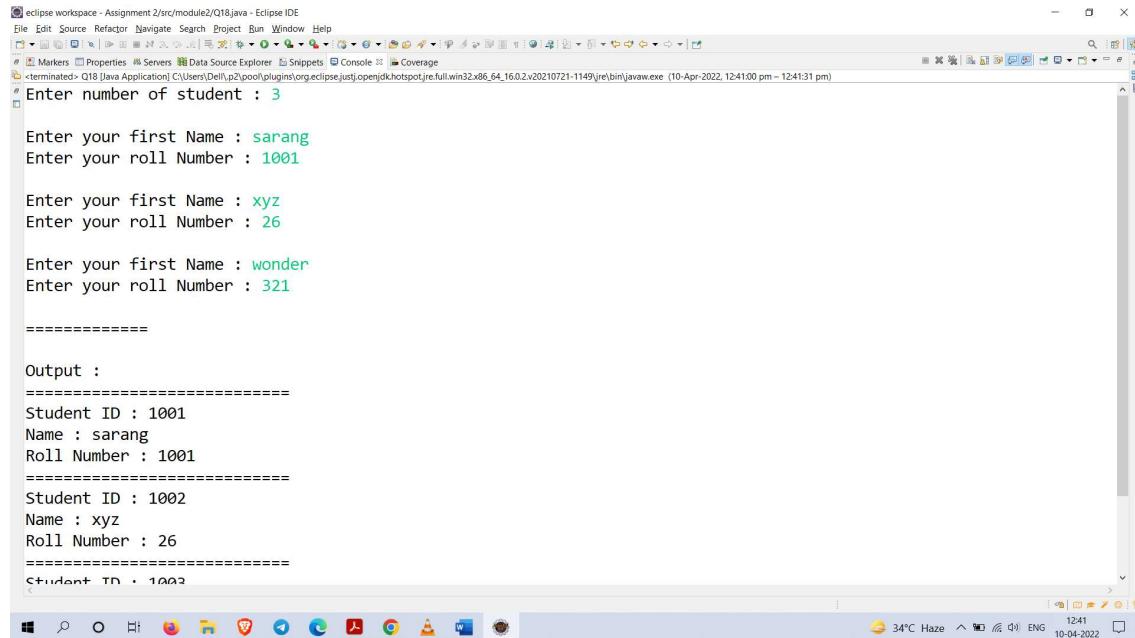
Code :

```
package module2;

import java.util.Scanner;

public class Q18 {
    private int sid;
    private String firstName;
    private int rollNumber;
    static int count=1000;
    Q18(String x, int y){
        count++;
        sid = count; //auto-generated sid
        this.firstName = x;
        this.rollNumber = y;
    }
    public void outputInfo() {
        System.out.println("Student ID : "+sid);
        System.out.println("Name : "+firstName);
        System.out.println("Roll Number : "+rollNumber);
        System.out.println("=====");
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter number of student : ");
        int n = sc.nextInt();
        System.out.println();
        Q18 o[] = new Q18[n];
        for(int i = 0 ; i < n ; i++) {
            System.out.print("Enter your first Name : ");
            String x = sc.next();
            System.out.print("Enter your roll Number : ");
            int y = sc.nextInt();
            System.out.println();
            o[i] = new Q18(x,y);
        }
        System.out.println("\nOutput : ");
        System.out.println("=====");
        for(Q18 z : o) {
            z.outputInfo();
        }
        sc.close();
    }
}
```

Output :



eclipse workspace - Assignment 2/src/module2/Q18java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Markers Properties Servers Data Source Explorer Snippets Console Coverage
terminated: Q18 Java Application C:\Users\Deft.p2\pool\plugins\org.eclipse.jst\openjdk.hotspot.jre.full.win32.x86_64_16.0.2.v20210721-1149\jre\bin\javaw.exe (10-Apr-2022, 12:41:00 pm - 12:41:31 pm)
Enter number of student : 3

Enter your first Name : sarang
Enter your roll Number : 1001

Enter your first Name : xyz
Enter your roll Number : 26

Enter your first Name : wonder
Enter your roll Number : 321

=====
Output :
=====

```
=====
Student ID : 1001
Name : sarang
Roll Number : 1001
=====
Student ID : 1002
Name : xyz
Roll Number : 26
=====
Student ID : 1003
```

Windows taskbar at the bottom: 34°C Haze, ENG, 12:41, 10-04-2022.

Q19. Design a Student class which contains register number, name, marks for three courses, average and result. Define methods to get input, print details, find the average and result (if mark is more than 49 in all courses then Pass, otherwise Fail). Keep input methods and print methods as public while others are private. Create array of objects for the Student class and demonstrate those methods.

Code :

```
package module2;

import java.util.Scanner;

public class Student {
    static Scanner sc = new Scanner(System.in);
    private int registerNumber;
    private String name;
    private float m1,m2,m3;
    private float average;
    private static int count;
    private static boolean value;
    private float sum;
    public void input() {
        System.out.print("Enter registerNumber : ");
        registerNumber = sc.nextInt();
        System.out.print("Enter Name : ");
        name = sc.next();
        value = false;
        while(value != true) {
            System.out.print("Enter Marks for subject 1 : ");
            m1 = sc.nextInt();
            System.out.print("Enter Marks for subject 2 : ");
            m2 = sc.nextInt();
            System.out.print("Enter Marks for subject 3 : ");
            m3 = sc.nextInt();
            if(m1 > 100 || m2 > 100 || m3 > 100 || m1 < 0 || m2 < 0
|| m3 < 0)
                value = false;
            else
                value = true;
            if(value == false)
                System.out.println("Invalid\nPlease try again...");
        }
    }
    private void process() {
        sum = m1 + m2 + m3;
        average = (sum)/3;
    }
}
```

```
public void output() {
    count++;
    System.out.print(count+" "+registerNumber+" "+name);
    System.out.print(" "+sum+" "+average+" ");
    if(m1 > 49 && m2 > 49 && m3 > 49)
        System.out.print("Pass");
    else
        System.out.print("Fail");
    System.out.println();
}
public static void main(String[] args) {
    System.out.println("Enter the number records to be filled : ");
    int n = sc.nextInt();
    Student s[] = new Student[n];
    for(int i = 0 ; i < n ; i++) {
        s[i] = new Student();
        System.out.println("\nData for student "+(i+1));
        s[i].input();
        s[i].process();
    }
    System.out.println("Record : ");
    for(Student l : s) {
        l.output();
    }
    sc.close();
}
```

Output :

```
eclipse workspace - Assignment 2/src/module2/Student.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Markers Properties Servers Data Source Explorer Snippets Console Coverage
terminated> Student [Java Application] C:\Users\Delhi.p2\pool\plugins\org.eclipse.jdt.core\1.14.0.v20210721-1149\jre\bin\javaw.exe (10-Apr-2022, 1:01:39 pm - 1:02:11 pm)
# Enter the number records to be filled :
2

Data for student 1
Enter registerNumber : 1000
Enter Name : sarang
Enter Marks for subject 1 : 49
Enter Marks for subject 2 : 49
Enter Marks for subject 3 : 49

Data for student 2
Enter registerNumber : 1002
Enter Name : deepak
Enter Marks for subject 1 : 50
Enter Marks for subject 2 : 50
Enter Marks for subject 3 : 50
Record :
1) 1000  sarang  147.0  49.0  Fail
2) 1002  deepak  150.0  50.0  Pass
```

Q20. Design a class to store account details of a person like account number, name, account type, available balance and minimum balance. Define methods to get input, display account details, show balance, deposit and withdraw. Apply the condition while withdraw money from account that the minimum balance to be maintained. Create a demo class in Java to demonstrate these methods with minimum of 3 objects.

Code :

```
package module2;

import java.util.Scanner;

public class Q20 {
    static Scanner sc = new Scanner(System.in);
    public int accNumber;
    public String name;
    public String accType;
    public float balance;
    public float minBalance;
    public float deposit;
    public float withdraw;

    public void setInput() {
        System.out.print("Account Number : ");
        accNumber = sc.nextInt();
        System.out.print("First Name : ");
        name = sc.next();
        System.out.print("Account Type : ");
        accType = sc.next();
        System.out.print("Balance : ");
        balance = sc.nextFloat();
        System.out.print("Minimum Balance : ");
        minBalance = sc.nextFloat();
    }

    public void getOutput() {
        System.out.println("=====");
        System.out.println("Name : "+name);
        System.out.println("Account Number : "+accNumber);
        System.out.println("Account type : "+accType);
        System.out.println("Current Balance : "+balance);
        System.out.println("=====\n");
    }

    public void showBalance() {
        System.out.println("\nCurrent Balance : "+balance);
    }

    public void deposit(){
        System.out.println("=====");
        System.out.println("Enter the amount to be deposit : ");
        deposit = sc.nextFloat();
        balance+=deposit;
        showBalance();
    }
}
```

```
public void withdraw() {
    System.out.println("=====");
    System.out.println("Enter the amount to be withdraw : ");
    withdraw = sc.nextFloat();
    if(withdraw > (balance - 2500)) {
        System.out.println("Account balance is insufficient.");
    }
    else
        balance-=withdraw;
    showBalance();
}
public static void main(String[] args) {
    Q20 a[] = new Q20[3];
    for(int i = 0 ; i < 3 ; i++)
    {
        a[i] = new Q20();
        a[i].setInput();
        a[i].deposit();
        a[i].withdraw();
        a[i].getOutput();
    }
    sc.close();
}
}
```

Output :

Part 1

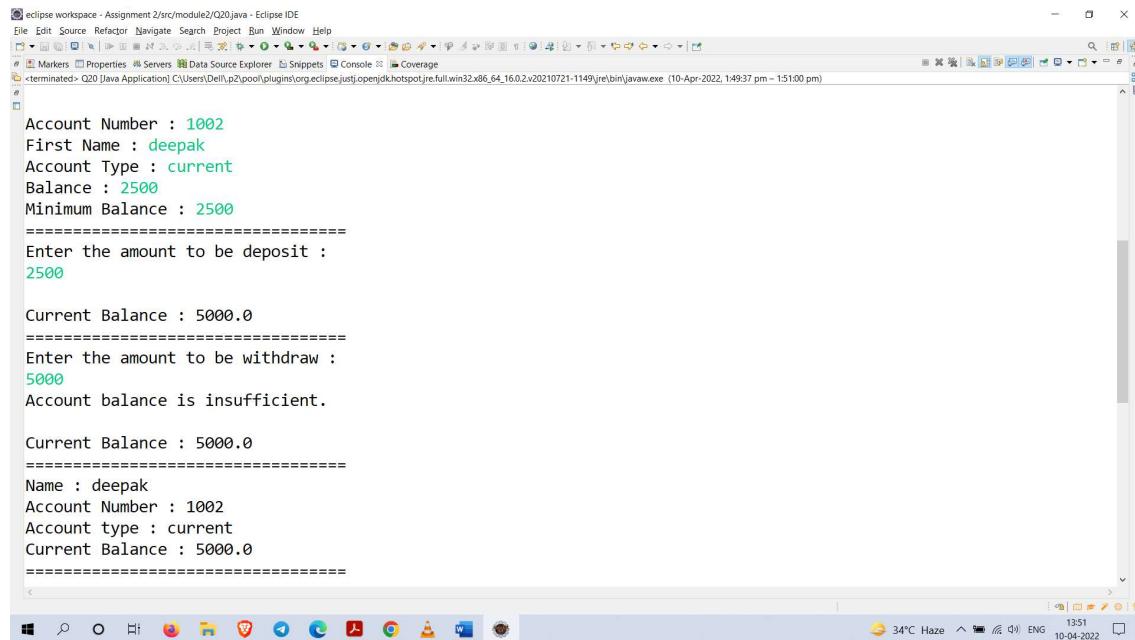
```
eclipse workspace - Assignment 2/src/module2/Q20.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Markers Properties Servers Data Source Explorer Snippets Console Coverage
terminated> Q20 [Java Application] C:\Users\DELL.p2pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_16.0.2.v20210721-1149\jre\bin\javaw.exe (10-Apr-2022, 1:49:37 pm - 1:51:00 pm)
Account Number : 1001
First Name : sarang
Account Type : saving
Balance : 10000
Minimum Balance : 2500
=====
Enter the amount to be deposit :
2500

Current Balance : 12500.0
=====
Enter the amount to be withdraw :
10000

Current Balance : 2500.0
=====
Name : sarang
Account Number : 1001
Account type : saving
Current Balance : 2500.0
=====

Account Number : 1002
First Name : deenak
```

Part 2



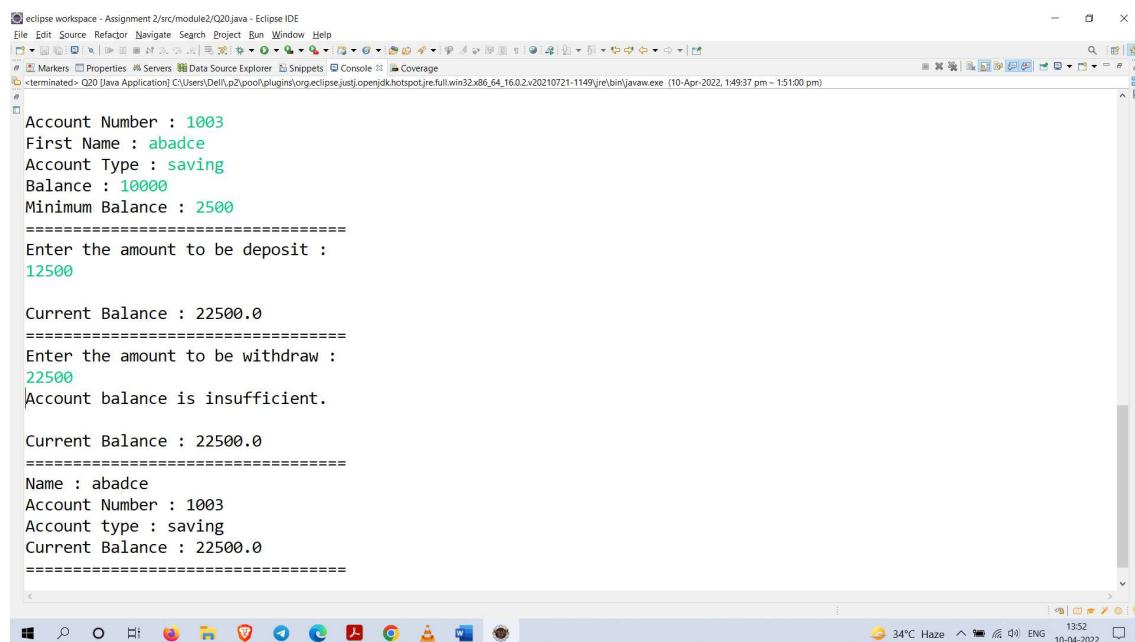
eclipse workspace - Assignment 2/src/module2/Q20.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Markers Properties Servers Data Source Explorer Snippets Console Coverage
<terminated> Q20 [Java Application] C:\Users\Deepl.p2\pool\plugins\org.eclipse.jst\openjdk.hotspot.jre.full.win32.x86_64_16.0.2.v20210721-1149\jre\bin\javaw.exe (10-Apr-2022, 1:49:37 pm - 1:51:00 pm)

Account Number : 1002
First Name : deepak
Account Type : current
Balance : 2500
Minimum Balance : 2500
=====
Enter the amount to be deposit :
2500

Current Balance : 5000.0
=====
Enter the amount to be withdraw :
5000
Account balance is insufficient.

Current Balance : 5000.0
=====
Name : deepak
Account Number : 1002
Account type : current
Current Balance : 5000.0
=====

Part 3



eclipse workspace - Assignment 2/src/module2/Q20.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Markers Properties Servers Data Source Explorer Snippets Console Coverage
<terminated> Q20 [Java Application] C:\Users\Deepl.p2\pool\plugins\org.eclipse.jst\openjdk.hotspot.jre.full.win32.x86_64_16.0.2.v20210721-1149\jre\bin\javaw.exe (10-Apr-2022, 1:49:37 pm - 1:51:00 pm)

Account Number : 1003
First Name : abadce
Account Type : saving
Balance : 10000
Minimum Balance : 2500
=====
Enter the amount to be deposit :
12500

Current Balance : 22500.0
=====
Enter the amount to be withdraw :
22500
Account balance is insufficient.

Current Balance : 22500.0
=====
Name : abadce
Account Number : 1003
Account type : saving
Current Balance : 22500.0
=====

Q22. Using loop statement write a program that prompts the user to enter 5 integer values:

- I. Find and display the Largest and Smallest number
- II. Display whether the number is Even or Odd
- III. Display whether the number is negative, positive or zero
- IV. Calculate the Sum and Average of the Even numbers

Code :

```
package module2;

import java.util.Scanner;

public class Q22 {
    static int largest;
    static int smallest;
    static int input[] = new int[5];
    static Scanner sc = new Scanner(System.in);
    static int even;
    static int odd;
    static int sum;
    public void input() {
        System.out.println("Enter 5 integers (hit enter after each
input): ");
        for(int i = 0 ; i < 5 ; i++) {
            System.out.print(">");
            input[i] = sc.nextInt();
        }
    }
    public void largeSmall() {
        largest = input[0];
        smallest = input[0];
        System.out.println("\nLargest and Smallest : ");
        System.out.println("=====");
        for(int i = 1 ; i < input.length ; i++)
        {
            if(input[i] > largest)
                largest = input[i];
            if(input[i] < smallest)
                smallest = input[i];
        }
        System.out.println("Largest Number is : "+largest);
        System.out.println("Smallest Number is : "+smallest);
    }
}
```

```
public void evenOdd() {
    System.out.println("\nEven and Odd : ");
    System.out.println("=====");
    for(int i = 0 ; i < input.length; i++) {
        if(input[i] % 2 == 0) {
            even = input[i];
            System.out.println(input[i]+" is even");
            sum+=even;
        }
        else {
            odd = input[i];
            System.out.println(input[i]+" is odd");
        }
    }
}
public void negiPosi() {
    System.out.println("\nNegative, Positive and Zero : ");
    System.out.println("=====");
    for(int i = 0 ; i < input.length ; i++) {
        if(input[i] < 0) {
            System.out.println(input[i]+" is Negative");
        }else if(input[i] > 0) {
            System.out.println(input[i]+" is Positive");
        }else if(input[i] == 0) {
            System.out.println(input[i]+" is Zero");
        }
    }
}
public void sumAvg() {
    System.out.println("\nSum and Average : ");
    System.out.println("=====");
    System.out.println("Sum of even numbers : "+sum);
    System.out.println("Average of even number : "+(sum/3));
}
public static void main(String[] args) {
    Q22 q = new Q22();
    q.input();
    q.largeSmall();
    q.evenOdd();
    q.negiPosi();
    q.sumAvg();
    sc.close();
}
}
```

Output :

```
eclipse workspace - Assignment 2/src/module2/Q22.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
File Edit Source Refactor Navigate Search Project Run Window Help
Markers Properties Servers Data Source Explorer Scripts Console Coverage
terminated: Q22 Java Application C:\Users\Devi\Downloads\plugins\org.eclipse.jdt.core\6.10.0\jre\bin\java.exe (10-Apr-2022, 1:57:09 pm - 1:57:17 pm)
ENTER 5 integers (will enter after each input):
=>1
=>2
=>3
=>4
=>55
Largest and Smallest :
Largest Number is : 55
Smallest Number is : 1
Even and Odd :
1 is odd
2 is even
3 is odd
4 is even
55 is odd
Negative, Positive and Zero :
1 is Positive
2 is Positive
3 is Positive
4 is Positive
55 is Positive
Sum and Average :
Sum of even numbers : 6
Average of even number : 2
```

Q23. Write a program that randomly generates an integer between 0 and 100, inclusive. The program prompts the user to enter a number continuously until the number matches the randomly generated number. For each user input, the program tells the user whether the input is too low or too high, so the user can choose the next input intelligently.

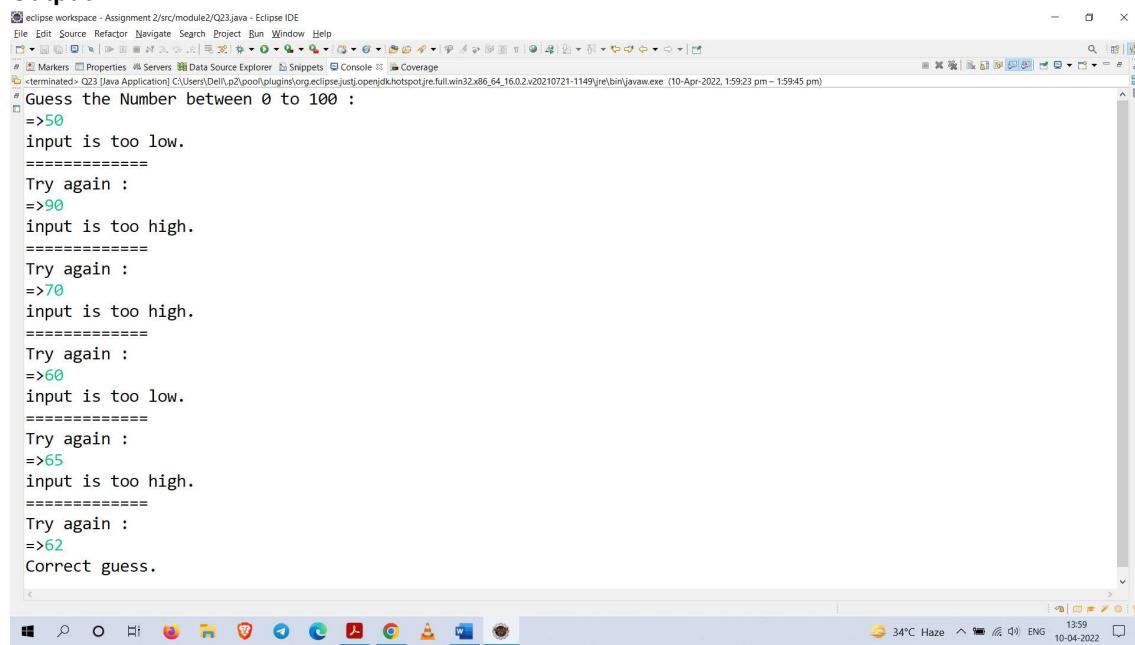
Code :

```
package module2;

import java.util.Scanner;

public class Q23 {
    static int guess;
    static int input;
    static Scanner sc = new Scanner(System.in);
    public static void main(String[] args) {
        guess = (int) (Math.random() * 101);
        System.out.println("Guess the Number between 0 to 100 :");
        System.out.print("=>");
        input = sc.nextInt();
        while(input != guess)
        {
            if(input < guess) {
                System.out.println("input is too low.");
            }else if(input > guess) {
                System.out.println("input is too high.");
            }
            System.out.println("=====");
            System.out.println("Try again : ");
            System.out.print("=>");
            input = sc.nextInt();
        }
        if(input == guess) {
            System.out.println("Correct guess.");
        }
        sc.close();
    }
}
```

Output :



The screenshot shows the Eclipse IDE interface with the title "eclipse workspace - Assignment 2/src/module2/Q23.java - Eclipse IDE". The Console view displays the output of a Java application named "Q23". The application is a猜数游戏 (guessing game) that asks the user to guess a number between 0 and 100. The user's input is shown in green, and the program's response is in black. The game continues until the correct guess is made.

```
# Guess the Number between 0 to 100 :
=>50
input is too low.
=====
Try again :
=>90
input is too high.
=====
Try again :
=>70
input is too high.
=====
Try again :
=>60
input is too low.
=====
Try again :
=>65
input is too high.
=====
Try again :
=>62
Correct guess.
```

Q24. Write a class named Duplicate that use a method named..... long duplicateDigits(long n) that, given a long integer n that is here guaranteed to be positive, returns the integer constructed by writing its digits twice in a row.

For example, when called with argument 1504, this method would return 15041504. You may not convert n to a string at any time, but must calculate the answer using only loops, conditions and basic integer arithmetic.

Your method must work correctly for all positive values of n less than one million.

(Hint: think of how you would check whether n consists of one digit, and how you would calculate the duplicated number in such case. Then think of two digits, three digits, four digits and higher until you can see and can implement the general pattern.)

Code :

```
package module2;

import java.util.Scanner;

public class Duplicate {
    public long duplicateDigits(long n) {
        long num = n;
        int count = 0, multiple=1;
        while(num>0) {
            num/=10;
            count++;
        }
        while(count!=0) {
            multiple*=10;
            count--;
        }
        num=(n*multiple)+n;
        return (int)num;
    }
    public static void main(String[] args) {
        long n=-1;
        Scanner sc = new Scanner(System.in);
        while(n<=0) {
            System.out.println("Enter positive number : ");
            n = sc.nextLong();
        }
        Duplicate dup = new Duplicate();
        int result = (int)dup.duplicateDigits(n);
        System.out.println("Result : "+result);
        sc.close();
    }
}
```

Output :

The screenshot shows the Eclipse IDE interface with a Java application running. The application's code is as follows:

```
# Enter positive number :
-23
Enter positive number :
23
Result : 2323
```

The application prompts the user to enter a positive number. The user enters "-23", which is then displayed as "Result : 2323". This indicates a bug in the application logic where negative numbers are being processed as if they were positive.

- Q25.** **Computing Mean and Standard Deviation:** Write a java program that computes the standard deviation of numbers. You may use a different but equivalent formula to compute the standard deviation of n numbers.

$$\text{Mean} = \frac{\sum_{i=1}^n x_i}{n} = \frac{x_1 + x_2 + x_3 + \dots + x_n}{n}$$

where x_i are the individual values of data
 n is the sample size or number of values

Formula to estimate sample standard deviation

$$\text{deviation} = \sqrt{\frac{\sum_{i=1}^n (x_i - \text{Mean})^2}{n - 1}}$$

Program should prompt the user to enter ten numbers and displays the mean and standard deviation, as shown in the following sample run:

```
Enter ten numbers: 2.9 1.5 2 6 5 3.3 4.2 7.7 5 4
The mean is :
The standard deviation is:
```

Code :

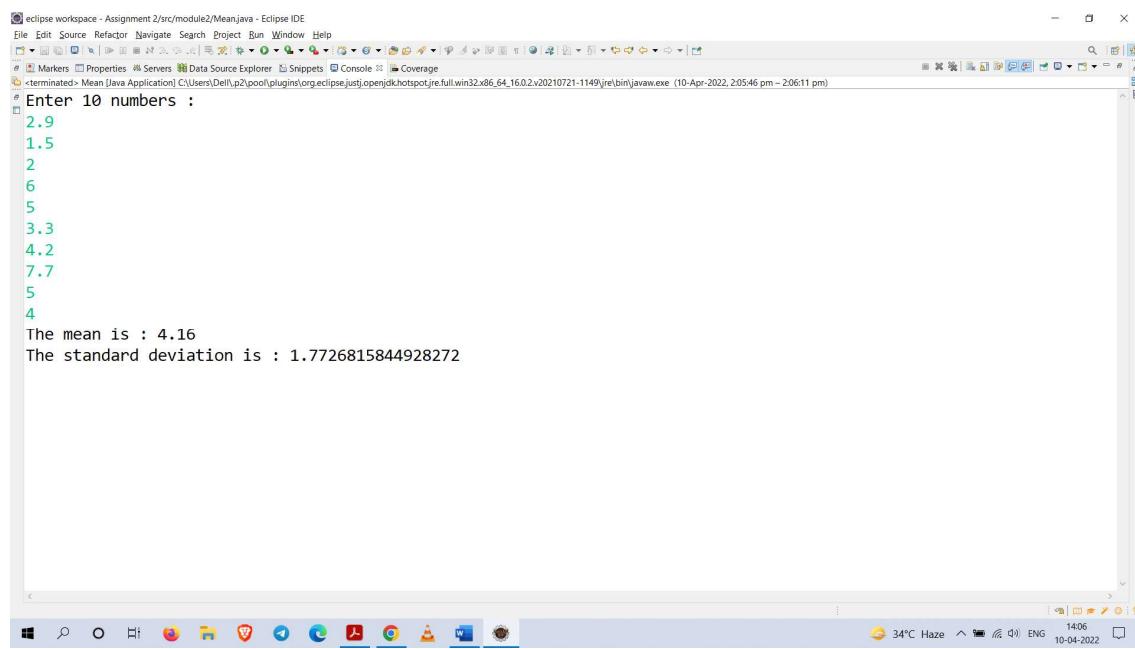
```
package module2;

import java.util.Scanner;

public class Mean {
    static Scanner sc = new Scanner(System.in);
    static int n;
    static double x[];
    static double sum;
    static double mean;
    public void input() {
        System.out.println("Enter 10 numbers : ");
        x = new double[10];
        for(int i = 0 ; i < x.length ; i++) {
            x[i] = sc.nextDouble();
        }
        n = x.length;
    }
    public void mean() {
        for(int i = 0 ; i < x.length ; i++) {
            sum+=x[i];
        }
        mean = sum / n;
        System.out.println("The mean is : "+mean);
    }
}
```

```
public void deviation() {
    double numerator = 0;
    for(int i = 0 ; i < x.length ; i++) {
        numerator += ((x[i] - mean) * (x[i] - mean));
    }
    double deviation = Math.sqrt(numerator/(n));
    System.out.println("The standard deviation is : "+deviation);
}
public static void main(String[] args) {
    Mean m = new Mean();
    m.input();
    m.mean();
    m.deviation();
    sc.close();
}
}
```

Output :



The screenshot shows the Eclipse IDE interface with a Java application named 'Mean' running. The code in the editor calculates the mean and standard deviation of a list of numbers entered by the user.

```
eclipse workspace - Assignment 2/src/module2/Mean.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Markers Properties Servers Data Source Explorer Snippets Console Coverage
<terminated> Mean [Java Application] C:\Users\Devi\p2\pool\plugins\org.eclipse.jdt.openjdk.hotspot.jre.full.win32.x86_64_16.0.2.v20210721-1149\jre\bin\javaw.exe (10-Apr-2022, 20546 pm - 20611 pm)
# Enter 10 numbers :
2.9
1.5
2
6
5
3.3
4.2
7.7
5
4
The mean is : 4.16
The standard deviation is : 1.7726815844928272
```

The status bar at the bottom right indicates the system temperature is 34°C, the time is 14:06, and the date is 10-04-2022.