EDUCATION

Professional Tecnology Education

IT

1) Create double variables for the prices of any 3 items. Print the sum of the prices on the console with a label.

```
double shirtPrice = 12.99;
double hatPrice = 23.99;
double shoesPrice = 15.99;
System.out.println(“Total price: “ + (shirtPrice + hatPrice + shoesPrice));
```

2) Create a float variable, a long variable, and an integer variable for any 3 items. Print the multiplication of the values on the console with a label.

```
float x = 15.99F;
long y = 13L;
int z = 12;
System.out.println("Multiplication: " + x*y*z);
```

3) Type a code to find simple interest.

Note: Simple interest formula = principal * rate * numberOfYear / 100

```
int principal = 10000, rate = 6, numberOfYear = 3;
int simpleInterest = principal * rate * numberOfYear/100;
System.out.println("The simple interest is " + simpleInterest);
```

4) Create a String and two Long variables. Print the sum and the multiplication of the long variables with the String on the console.

```
String s = "The result is ";
Long a = 123L;
Long b = 9123456782L;
System.out.println(s + (a+b));
System.out.println(s + a*b);
```

5) **Create two boolean variables whose values are different and print their values in the same line with a space between two consecutive values.**

```
boolean a = true;
boolean b = false;
System.out.println(a + " " + b);
```

6) **Create 3 float variables for the price of a book, notebook, and laptop.**

Print the total price of 2 books, 4 notebooks and 3 laptops on the console

```
float book = 12.99F, noteBook = 23.45F, laptop = 34.12F;
float totalPrice = 2*book + 4*noteBook + 3*laptop;
System.out.println("The total price is " + totalPrice);
```

7) **Type a code to swap two integers.**

```
Scanner scan = new Scanner(System.in);
System.out.println("Enter 2 numbers to swap");
double n1 = scan.nextDouble();
double n2 = scan.nextDouble();
System.out.println("Before swapping: " + n1 + " - " + n2);
```

1. **Way: Use 3rd variable**

```
double temp = n1;
n1 = n2;
n2 = temp;
System.out.println("After swapping: " + n1 + " - " + n2);
```

2. **Way: Do not use 3rd variable**

```
n1 = n1 + n2;
n2 = n1 - n2;
n1 = n1 - n2;
System.out.println("After swapping: " + n1 + " - " + n2);
```

1) **Create a short variable and convert it to an int variable.**

```
short age = 12;
int newAge = age;
```

2) **Create a long variable and convert it to an int variable.**

```
long age = 12L;
int newAge = (int)age;
```

3) **Create a double variable and convert it to a float variable**

```
double price = 2.34;
float newPrice = (float)price;
```

4) **Create a double variable and convert it to a short variable, then print the value of the short variable on the console. Be careful about the output, it will not be a decimal value.**

```
double price = 1.2;
short newPrice = (short)price;
System.out.println(newPrice);
```

5) **Create a byte variable and convert it to a double variable, then print the value of the double variable on the console. Be careful about the output, it will be a decimal number.**

```
byte age = 23;
double newAge = age;
System.out.println(newAge);
```

1) **Type a code to find the sum of the minimum value of the byte and the maximum value of short data types.**

```
byte min = Byte.MIN_VALUE;
short max = Short.MAX_VALUE;
System.out.println(min + max);
```

2) **Type a code to convert** “103” **String to byte and to convert** “2351” **String to short then print the difference on the console.**

```
byte first = Byte.valueOf("113");
short second = Short.valueOf("2351");
System.out.println(second - first);
```

3) **What are the differences between valueOf(String s) and parseByte(String s) methods?**

parseByte() returns primitive integer type (int), whereas valueOf() returns java.lang.Integer, which is the object representative of the integer. There are circumstances where you might want an Integer object, instead of a primitive type.

Of course, another obvious difference is that valueOf() is an instance method whereas parseByte() is a static method.

1) **Create a String variable for city names which have just a single word. Print the city name with the initial is in uppercase and all the other characters** are in lower cases on the console.

Example: mIAMI should be printed as Miami
miami should be printed as Miami
MIAMI should be printed as Miami
mlaMi should be printed as Miami etc.

```
String cityName = "  mIAMI  ";
String updatedCityName = cityName.trim().toLowerCase();
updatedCityName = updatedCityName.substring(0,1).toUpperCase() + updatedCityName.substring(1);
System.out.println(updatedCityName);
```

2) **Create 3 String variables for people's names. Print the sum of the number of characters in all the 3 names except space characters.**
Example: If the names are Ali Can, Merve Star, Mark Tom you should see 22 on the console.

```
String name1 = "Ali Can";
String name2 = "Aliye Canan";
String name3 = "Aliyev Can Cananov";
Integer c1 = name1.replaceAll("\\s","").length();
Integer c2 = name2.replaceAll("\\s","").length();
Integer c3 = name3.replaceAll("\\s","").length();
System.out.println("Total number of characters different from space: " + (c1 + c2 + c3));
```

3) **Create a String variable, print the total number of alphabetical and numeric characters in the String on the console.**
Example: If the String is '   Miami 33018!!!   ' you need to print 10 on the console.

```
String s = "   Miami 33018!!!   ";
Integer numOfChars = s.replaceAll("\\s","").replaceAll("[⊥A-Za-z0-9]","").length();
System.out.println("Total num of alphabetical and numeric characters chars:" + numOfChars);
```

4) **Create a String variable, print the number of non-digit characters in the String on the console.**
Example: If the String is '1a3Bcf4!...' you need to print 8 on the console.

```
String s = "2a3B4?-!5";
Integer numOfNonDigitChars = s.replaceAll("[0-9]","").length();
System.out.println("Total number of non-digit characters: " + numOfNonDigitChars);
```

5) **Create a String variable and print just the last non-space character on the console for any String.**
Example: For 'Ali Can' you should print n
For 'Miami   ' you should print i etc.

```
String s = "Miami      ";
int idxOfLastNonSpace = s.trim().length()-1;
String lastNonSpaceChar = s.substring(idxOfLastNonSpace, idxOfLastNonSpace+1);
System.out.println("The non-space last character: " + lastNonSpaceChar);
```

6) Create a String variable and find the sum of the ASCII values of the first and the last characters of the String.

```
String s = "Miami";
int indexOfLastChar = s.length()-1;
int AsciiOfFirst = s.charAt(0);
int AsciiOfLast = s.charAt(indexOfLastChar);
System.out.println("Total ASCII values of first and last chars: " + (AsciiOfFirst + AsciiOfLast));
```

7) Create a String variable and print all characters except the first character on the console.
Example:If the String is 'Java' you should print 'ava' on the console.

```
String s = "Java";
String allCharsExceptFirstChar = s.substring(1);
System.out.println("All characters except the first character: " + allCharsExceptFirstChar);
```

8) Create a String variable and print all characters except the last character on the console in uppercases.
Example:If the String is 'Java' you should print 'JAV' on the console.

```
String s = "Java";
Integer indexOfLastChar = s.length()-1;
String allCharsExceptLastChar = s.substring(0, indexOfLastChar).toUpperCase();
System.out.println("All characters except the last character: " + allCharsExceptLastChar);
```

9) Create a String variable and print all characters except the first character and the last character on the console in uppercases.
Example:If the String is 'Java' you should print 'AV' on the console.

```
String s = "Java";
Integer indexOfLastChar = s.length()-1;
String allCharsExceptLastCharInUpperCase = s.substring(1, indexOfLastChar).toUpperCase();
System.out.println("All characters except the first and the last character: " + allCharsExceptLastCharInUpperCase);
```

10) Type code to check if a String has just a single space character any position in the middle
Example: For 'Ali   Can' your code should print false on the console
For '   Ali Can   ' your code should print false on the console
For '   Ali   Can   ' your code should print false on the console
For 'Ali Can' your code should print true on the console

```
String s = "Tom Hanks";
String trimmed = s.trim();
String allExceptSpace = trimmed.replaceAll("\\s", "");
Boolean isThereSingleSpaceInTheMiddle = trimmed.length() - allExceptSpace.length()==1;
System.out.println("There is a space in the middle: " + isThereSingleSpaceInTheMiddle);
```

11) Type code to check if a String does not have any space character at the beginning and at the end?
Example: For '   Ali   ' your code should print false on the console
For 'Ali' your code should print true on the console

```
String s = "  Tom ";
String trimmedString = s.trim();
Boolean result  = s.equals(trimmedString);
System.out.println("Is there space at the beginning and/or at the end? " + !result);
```

12) Type code to check if a String has an uppercase initial and dot at the end.

Example: For ‘Ali’ your code should print false on the console

For ‘ali.’ your code should print false on the console

For ‘   Ali.   ’ your code should print false on the console

For ‘Ali.’ your code should print true on the console

For ‘ALI.’ your code should print true on the console

```
String s = "Tom Hanks.";
char firstChar = s.charAt(0);
char lastChar = s.charAt(s.length()-1);
Boolean isTheFirstCharUpper = firstChar>='A' && lastChar<='Z';
Boolean isTheLastCharDot = lastChar=='.';
Boolean isFirstUpperAndLastDot = isTheFirstCharUpper && isTheLastCharDot;
System.out.println("Is the first char upper and the last char dot? " + isFirstUpperAndLastDot);
```

13) Type code to check if a password is valid or not for the following conditions;

Password must have at least 8 characters different from space character

Password must have at least 1 symbol

Example: For ‘A2b!’ your code should print false on the console

For ‘A2b3cdef’ your code should print false on the console

For ‘!1a23b4’ your code should print false on the console

For ‘!1a23b4?es’ your code should print true on the console

For ‘! a b 3 k’ your code should print false on the console

```
String pwd = "!1a23b4?es";
Boolean atLeastEightChars = pwd.replaceAll("\\s","").length()>=8;
Boolean atLeastOneSymbol = pwd.replaceAll("\\s","").replaceAll("[A-Za-z0-9]","").length()>0;
System.out.println("Is the password valid? " + (atLeastEightChars && atLeastOneSymbol));
```

14) **Type code to check if a String has a specific single character or not in three different ways.**

1.Way:

```
String s = "Java";
Boolean result = s.contains("v");
System.out.println("Does the character exist? " + result);
```

2.Way:

```
String s = "Java";
Boolean result = s.replaceAll("[⊥v]", "").length()>0;
System.out.println("Does the character exist? " + result);
```

2.Way:

```
String s = "Java";
Boolean result = s.indexOf("v")!=-1;
System.out.println("Does the character exist? " + result);
```

15) **String shirtPrice = '$12.99';**
String bookPrice = '$35.99';
Type code to find the sum of the shirt and book prices.

```
String s =  shirtPrice.replaceAll("[$.]","");
String b =  bookPrice.replaceAll("[$.]","");
double x = Double.valueOf(s);
double y = Double.valueOf(b);
System.out.println("Total price is " + (x+y)/100);
```

16) Type code to get initials of the first name and the last name of a given name. Middle name is out of scope.
Example: Tom Hanks ==> TH, Mary Star ==> MS

1.Way:

```
String name = "Tom Hanks";
String initialOfFirstName = name.substring(0,1);
int indexOfInitialOfLastName = name.indexOf(" ") + 1;
String initialOfLastName = name.substring(indexOfInitialOfLastName, indexOfInitialOfLastName+1);
System.out.println(initialOfFirstName + initialOfLastName);
```

2.Way:

```
String name = "Tom Hanks";
String initialOfFirstName = name.substring(0,1);
String initialOfLastName = name.split(" ")[1].substring(0, 1);
System.out.println(initialOfFirstName + initialOfLastName);
```

17) Type a code to find the number of punctuation marks used in a String.

```
String s = "Wow! Ali is 13 years old, but he is a university student.";
int totalNumberOfCharacters = s.length();
int charsDiffFromPunctuationMarks = s.replaceAll("\\p{Punct}", "").length();
int numOfPunctuationMarks = totalNumberOfCharacters - charsDiffFromPunctuationMarks;
System.out.println(numOfPunctuationMarks);
```

1) **Type the code that finds out how many days Ali lived.
Date of birth of Ali is 12th of May 2002**

```
LocalDate birthDateOfAli = LocalDate.of(2002, 5, 12);
LocalDate currentDate = LocalDate.now();
long numberOfDaysAliLived = ChronoUnit.DAYS.between(birthDateOfAli, currentDate);
System.out.println("The number of days Ali lived so far is " + numberOfDaysAliLived);
```

2) **Type the code that finds out how many months Ali lived.
Date of birth of Ali is 4th of June 1997**

```
LocalDate birthDateOfAli = LocalDate.of(1997, 06, 04);
LocalDate currentDate = LocalDate.now();
long numberOfMonthsAliLived = ChronoUnit.MONTHS.between(birthDateOfAli, currentDate);
System.out.println("The number of months Ali lived so far is " + numberOfMonthsAliLived);
```

3) **Date of birth of Ali is 4th of June 1997. Type code to find the exact date 2 years, 3 months, and 12 days after Ali's birth date.**

```
LocalDate birthDateOfAli = LocalDate.of(1997, 06, 04);
LocalDate exactDate = birthDateOfAli.plusYears(2).plusMonths(3).plusDays(12);
System.out.println("Exact date is " + exactDate);
```

4) **Ali was born 45 years, 8 months and 5 days after 29 October 1923.**
Veli was born 24 years, 2 months and 11 days before 15 September 1993.

Type code to calculate the exact date of birth of Ali and Veli

Type code to check if the date of birth of Ali and Veli is the same or not.

```
LocalDate dobAli = LocalDate.of(1923, 10, 29).plusYears(45).plusMonths(8).plusDays(5);
LocalDate dobVeli = LocalDate.of(1993, 9, 15).minusYears(24).minusMonths(2).minusDays(11);
System.out.println("Date of births are same " + dobAli.equals(dobVeli));
```

5) **Veli was born 3 years and 11 days after Ali.**
Ali gave you his date of birth as 24th of November 2012

Type code to calculate Veli’s date of birth.

```
LocalDate dobAli = LocalDate.parse("2012-11-24");
LocalDate dobVeli = dobAli.plusYears(3).plusDays(11);
System.out.println("The date of birth of Veli is " + dobVeli);
```

6) **Create a Date Value for your birth date and create Date Value for your kid's birth date then calculate the difference in days.**

```
LocalDate myDob = LocalDate.of(1986, 8, 21);
LocalDate myKidsDob = LocalDate.of(2015, 3, 12);
long difference = ChronoUnit.DAYS.between(myDob, myKidsDob);
System.out.println("Number of days between two dates is " + difference);
```

7) **Get the last 2 digits of the year in current date**

```
LocalDate myDob = LocalDate.of(1996, 8, 21);
int lastTwoDigits = myDob.getYear()%100;
System.out.println("Last 2 digits of the year is " + lastTwoDigits);
```

8) Type code to check if a given year is “Leap year” **or not**

Leap Year:

i) If a year is divisible by 100 and divisible by 400 it is called leap year. For example; 2000 is, 1900 is not

ii) If a year is not divisible by 100 and divisible by 4 it is called leap year. For example; 2004 is, 2007 is not

```
LocalDate myDob = LocalDate.of(1996, 8, 21);
boolean isLeap = myDob.isLeapYear();
System.out.println("The year is leap year " + isLeap);
```

9) Find the sum of the month numbers of two different dates

```
LocalDate myDob = LocalDate.of(1996, 8, 21);
LocalDate myKidsDob = LocalDate.of(2015, 3, 12);
System.out.println("The sum is " + myDob.getMonth() + myKidsDob.getMonth());
```

10) Find the difference in hours for two different dates

```
LocalDate myDob = LocalDate.of(1996, 8, 21);
LocalDate myKidsDob = LocalDate.of(2015, 3, 12);
long differenceInHours = ChronoUnit.HOURS.between(myDob, myKidsDob);
System.out.println("The difference in hours is " + differenceInHours);
```

1) If the hour is

i) Between 24:00 and 05:00 then print 'Sleeping time' on the console

ii) Between 08:00 and 16:00 then print 'Working time' on the console

iii) Between 19:00 and 22:00 then print 'Family time' on the console

iv) For the others print 'Personal time' on the console

```
LocalTime now = LocalTime.now();
int hour = now.getHour();

if(hour>0 && hour<5){
  System.out.println("Sleeping time");
}else if(hour>8 && hour<16){
  System.out.println("Working time");
}else if(hour>19 && hour<22){
  System.out.println("Family time");
}else{
  System.out.println("Personal time");
}
```

2) Type code to find time difference between Japan time zone and Germany time zone

```
LocalDateTime timeInJapan = LocalDateTime.now(ZoneId.of("Asia/Tokyo"));
System.out.println(timeInJapan);
LocalDateTime timeInGermany = LocalDateTime.now(ZoneId.of("Europe/Berlin"));
System.out.println(timeInGermany);
long difference = ChronoUnit.HOURS.between(timeInGermany, timeInJapan);
System.out.println(difference);
```

3) **Ali was born in Istanbul on 5th of February 2015 at 10:00 am TRT, Mark was born in the USA on 5th of February 2015 at 12:00 pm EST. What is the difference between Ali’s birth time and Mark’s birth time in hours?**

```
LocalDateTime dobAli = LocalDateTime.of(2015, 2, 5, 10, 0, 0);
ZonedDateTime localDobAli = dobAli.atZone(ZoneId.of("Europe/Istanbul"));
LocalDateTime dobVeli = LocalDateTime.of(2015, 2, 5, 12, 0, 0);
ZonedDateTime localDobVeli = dobVeli.atZone(ZoneId.of("America/New_York"));
long differenceInHours = ChronoUnit.HOURS.between(localDobAli, localDobVeli);
System.out.println("The difference in hours is " + differenceInHours);
```

4) **Ali was born on 5th of February 2015 at 10:15 am EST, Mark was born on 6th of March 2015 at 12:25 pm EST. Ali remained in intensive care for 23 hours and 35 minutes. Veli remained in intensive care for 1 hour and 12 minutes. What is the difference in minutes between the times of Ali and Veli leaving the intensive care unit?**

```
LocalDateTime dobAli = LocalDateTime.of(2015, 2, 5, 10, 15, 0);
LocalDateTime dobMark = LocalDateTime.of(2015, 3, 6, 12, 25, 0);
long differenceInHours = ChronoUnit.MINUTES.between(dobAli, dobMark);
System.out.println("The difference in hours is " + differenceInHours);
```

5) **Mark was born on 2/23/2018 at 3:25 PM in GMT. What is the exact birth date and time of Mark in Japan?**

```
LocalDateTime timeInTokyo = LocalDateTime.now(ZoneId.of("Asia/Tokyo"));
LocalDateTime timeInGmt = LocalDateTime.now(ZoneId.of("GMT"));
long diffInHours = ChronoUnit.HOURS.between(timeInGmt, timeInTokyo);
LocalDateTime dobMark = LocalDateTime.of(2018, 2, 23, 15, 25, 0).minusHours(diffInHours);
System.out.println("Date of birth in Tokyo is " + dobMark);
```

1) **Type code to print**

   a) "Winter" for December, January, February

   b) "Spring" for March, April, May

   c) "Summer" for June, July, August

   d) "Fall" for September, October, November

   e) "Invalid month name" for all the others

```
String mName = "December";
mName = mName.toLowerCase();
if(mName.equals("december") || mName.equals("january") || mName.equals("february")){
  System.out.println("Winter");
}else if(mName.equals("march") || mName.equals("april") || mName.equals("may")){
  System.out.println("Spring");
}else if(mName.equals("june") || mName.equals("july") || mName.equals("august")){
  System.out.println("Summer");
}else if(mName.equals("september") || mName.equals("october") || mName.equals("november")){
  System.out.println("Fall");
}else{
  System.out.println("Invalid month name");
}
```

2) **Type code to print**

a) "Valid Password" if the password has at least 8 characters different from space character

b) "Do not use space character in password" if the password has any space character in any position

c) "Invalid Password" if a and b conditions are not satisfied

Note: Be careful about the orders of conditions in the solution

```
String password = "a1b2c3d4";
if(password.replaceAll("\\S", "").length()>0){
  System.out.println("Do not use space character in password");
}else if(password.replaceAll("\\s", "").length()>=8){
  System.out.println("Valid Password");
}else{
  System.out.println("Invalid Password");
}
```

3) **Type code to print**

a) "Round up by last digit:" and print the rounded value if the last digit is greater than or equal to 5

b) "Round down by last digit" and print the rounded value if the last digit is less than 5

```
int i = 124;
if(i%10>=5){
  System.out.println("Round up by last digit:" + (i/10+1)*10);
}else{
  System.out.println("Round down by last digit:" + (i/10)*10);
}
```

4) **Type code to print the name of the month when you entered the number of the month. Example; For 1 output is "January", for 2 output is "February" etc.**

```
int numOfMonth = 3;
if(numOfMonth==1){
  System.out.println("January");
}else if(numOfMonth==2){
  System.out.println("February");
}else if(numOfMonth==3){
  System.out.println("March");
}else if(numOfMonth==4){
  System.out.println("April");
}else if(numOfMonth==5){
  System.out.println("May");
}else if(numOfMonth==6){
  System.out.println("June");
}else if(numOfMonth==7){
  System.out.println("July");
}else if(numOfMonth==8){
  System.out.println("August");
}else if(numOfMonth==9){
  System.out.println("September");
}else if(numOfMonth==10){
  System.out.println("November");
}else if(numOfMonth==11){
  System.out.println("October");
}else if(numOfMonth==12){
  System.out.println("December");
}else{
  System.out.println("Invalid month number");
}
```

5) **Type code to print**

a) "Isosceles Triangle" if two sides of a triangle are equal in length.

b) "Equilateral Triangle" if all sides of a triangle are equal in length.

c) "Neither isosceles, nor equilateral" if a and b conditions are not satisfied

```
int a = 5;
int b = 3;
int c = 4;

if (a == b && b == c) {
  System.out.println("Equilateral Triangle");
} else if (a == b && b != c || a == c && b != c || b == c && a != c) {
  System.out.println("Isosceles Triangle");
} else {
  System.out.println("Neither isosceles, nor equilateral");
}
```

6) **Type code to create a converter which converts mile to km, second to hour, fahrenheit to celsius. Find the formulas for conversions from Google.**

a) When user entered 10 and mileToKm String, your code should print "16 km"(The number will be dynamic) on the console

b) When user entered 7200 and secondsToHours String, your code should print "2 Hours"(The number will be dynamic) on the console

c) When user entered 83 and fahrenheitToCelsius String, your code should print "28.3333 celsius"(The number will be dynamic) on the console

```
double mile = 10;
double seconds = 7200;
double fahrenheit = 83;
String operator = "fahrenheitToCelsius";

if(operator.equals("mileToKm")){
  System.out.println(mile*1.60934);
}else if(operator.equals("secondsToHours")){
  System.out.println((seconds/60)/60);
}else if(operator.equals("fahrenheitToCelsius")){
  System.out.println((fahrenheit-32)*5/9);
}
```

7) Type code to check the grammatical rules for full name

a) Your code should print "Error in initials" for "ali Can", "Ali can", "ali can"

b) Your code should print "First name or last name missed" for single words like "Ali" or "Can" or "ali"

c) Your code should print "Format error" for all the format like "ALI CAN"

d) Your code should print "Name was not entered" for one or more space characters like " " or "   "

e) Your code should print "Invalid Name" if the name has any character different from letters and space.

Note: If the abbreviation has more than 1 error all related error messages should be printed. For example; for "ali3" your code should print "Error in initials", "First name or last name missed", and "Invalid Name"

```
String fullName = "ali Can?";

String initialOfFirstName = fullName.substring(0,1);
Integer indexOfSpace = fullName.trim().indexOf(' ');
String initialOfLastName = fullName.substring(indexOfSpace+1,indexOfSpace+2);
Boolean isFirstNameInitialUpperCase = (initialOfFirstName.charAt(0)>='A' && initialOfFirstName.
charAt(0)<='Z');
Boolean isLastNameInitialUpperCase = (initialOfLastName.charAt(0)>='A' && initialOfLastName.
charAt(0)<='Z');


if(indexOfSpace==-1){
  System.out.println("First name or last name missed");
}
if(!isFirstNameInitialUpperCase || !isLastNameInitialUpperCase){
  System.out.println("Error in initials");
}
if(fullName.equals(fullName.toUpperCase())){
  System.out.println("Format error");
}
if(fullName.replaceAll("\\s","").length()==0){
  System.out.println("Name was not entered");
}
if(fullName.replaceAll("\\s","").replaceAll("[A-Za-z]","").length()>0){
  System.out.println("Invalid Name");
}
```

8) Type code to check the format of state abbreviations in the USA

a) Your code should print "State abbreviations cannot have more than 2 characters" if the abbreviation has more than 2 characters

b) Your code should print "State abbreviations cannot have lowercase characters" if the abbreviation has lowercase characters

c) Your code should print "State abbreviations cannot have characters different from letters" if the abbreviation has characters different from letters.

Note: If the abbreviation has more than 1 error all related error messages should be printed. For example; for "fl3" your code should print, "State abbreviations cannot have more than 2 characters", "State abbreviations cannot have lowercase characters", and "State abbreviations cannot have characters different from letters"

```
String stateAbbreviation = "Fl3";
if(stateAbbreviation.length()>2){
  System.out.println("State abbreviations cannot have more than 2 characters");
}
if(stateAbbreviation.replaceAll("[⊥a-z]","").length()!=0){
  System.out.println("State abbreviations cannot have lowercase characters");
}
if(stateAbbreviation.replaceAll("[⊥A-Za-z]","").length()!=0){
  System.out.println("State abbreviations cannot have characters different from letters");
}
```

9) **Type code to create a simple calculator which does addition, subtraction, multiplication, and division with any 2 number**

a) When user entered 10.2 and 5 and + sign your code should print "The result is 15.2" on the console

b) When user entered 10 and 5 and - sign your code should print "The result is 5.0" on the console

c) When user entered 10 and 5.3 and * sign your code should print "The result is 53.0" on the console

d) When user entered 10 and -5 and / sign your code should print "The result is -2.0" on the console

```
double a = 10.2,  b=5;
String operator = "+";
if(operator.equals("+")){
  System.out.println(a+b);
}else if(operator.equals("-")){
  System.out.println(a-b);
}else if(operator.equals("*")){
  System.out.println(a*b);
}else if(operator.equals("/")){
  System.out.println(a/b);
}
```

10) Type code to print the messages about the Body Mass Index(BMI) for the given BMI values

Invalid BMI value < 0

Underweight = <18.5

Normal weight = 18.5–24.9

Overweight = 25–29.9

Obesity = BMI of 30 or greater

```
double bmi = 12;
if(bmi<0){
  System.out.println("Invalid BMI value");
}else if(bmi<18.5){
  System.out.println("Underweight");
}else if(bmi<24.9 && bmi>=18.5){
  System.out.println("Normal weight");
}else if(bmi<29.9 && bmi>=25){
  System.out.println("Overweight");
}else if(bmi>30){
  System.out.println("Obesity");
}
```

1) **Use ternary to print "Valid Password" if the password has at least 8 characters different from space characters. "Invalid Password" if the password has less than 8 characters different from space character**

```
String password = "a1b 2c3";
String isValid = password.replaceAll("\\s","").length()>7 ? "Valid Password" : "Invalid Password";
System.out.println(isValid);
```

2) **Use ternary to print**

a) "Isosceles Triangle" if two sides of a triangle are equal in length.

b) "Equilateral Triangle" if all sides of a triangle are equal in length.

c) "Neither isosceles, nor equilateral" if a and b conditions are not satisfied

```
int a = 5, b = 4, c = 3;
String typeOfTriangle = a==b && b==c ? "Equilateral Triangle" : (a==b && b!=c || a==c && b!=c ||
b==c && a!=c ? "Isosceles Triangle" : "Neither isosceles, nor equilateral");
System.out.println(typeOfTriangle);
```

3) **Use ternary to print**

a) "Round up by last digit" and print the rounded value if the last digit of an integer is greater than or equal to 5

b) "Round down by last digit" and print the rounded value if the last digit of an integer is less than 5

For example; **127 will be rounded up and the value will be 130**

125 will be rounded up and the value will be 130

123 will be rounded down and the value will be 120

```
int i = 125;
String result = i%10>=5 ? ("Round up:" + (i/10+1)*10) : ("Round down:" + (i/10)*10);
System.out.println(result);
```

4) Type Apex code by using nested ternary.

Write a program to check if a year is a leap year or not.

If the year is divisible by 100 then it must be divisible by 400.

If a year is not divisible by 100 then it must be divisible by 4.

```
int y = 1800;
String isLeap = (y%100==0) ? ( (y%400==0) ? ("Leap") : ("Not Leap") ) : ( (y%4==0 ? ("Leap") : ("Not
Leap") ));
System.out.println(isLeap);
```

5) Type code to check the password

If it has more than 8 characters, initial should be 'i'

If it does not have more than 8 characters initial should be 'K'

Solve the task by using nested-ternary

```
String p = "XXXXXXXXX";
String r = p.length()>8 ? p.startsWith("i") ? "Valid" : "Invalid" : p.startsWith("K") ? "Valid" : "Invalid";
System.out.println(r);
```

6) Type code to calculate the absolute value of a number

For positive numbers and zero absolute value is the same with the number

For negative numbers to find absolute value multiply the number by -1

```
int i = -5;
int result = i<0 ? -1*i : i;
System.out.println(result);
```

7) Write a program to print the minimum of 2 integers on the console by using ternary.

```
int i = 13, k = 12;
int result = i<k ? i : k;
System.out.println(result);
```

8) **If the number has 3 digits, the output will be** “This number has 3 digits.” **Otherwise, the output will be** “This number has no 3 digits.”

1.Way:

```
int i = 750;
String r = (i>99 && i<1000) || (i>-1000 && i<-99) ? "Has 3 digits" : "Has no 3 digits";
System.out.println(r);
```

2.Way:

```
int i = -750;
i = Math.abs(i);
String r = (i>99 && i<1000) ? "Has 3 digits" : "Has no 3 digits";
System.out.println(r);
```

9) **Print 'Even' for even integers, print ‘Odd’ for odd integers by using ternary**

```
int i = 14;
String result = i%2==0 ? "Even" : "Odd";
System.out.println(result);
```

10) **If the number is positive print ‘Positive’, otherwise print ‘Not positive’ on the console by using ternary.**

```
int i = -12;
String result =  i>0  ? "Positive" : "Not positive";
System.out.println(result);
```

1) **Use switch statement to print**

a) "Winter" for December, January, February

b) "Spring" for March, April, May

c) "Summer" for June, July, August

d) "Fall" for September, October, November

e) "Invalid month name" for all the others

```
String monthName = "June";
monthName = monthName.toLowerCase();
switch(monthName){
  case "december":
  case "january":
  case "february":
     System.out.println("Winter");
     break;
  case "march":
  case "april":
  case "may":
     System.out.println("Spring");
     break;
  case "june":
  case "july":
  case "august":
     System.out.println("Summer");
     break;
  case "september":
  case "november":
  case "october":
     System.out.println("Fall");
     break;
  default:
     System.out.println("Invalid month name");
     break;
}
```

2) **Use a switch statement to print the name of the month when you enter the number of the month. For example; if user enters 1 your code should print "January",**
if user enters 2 your code should print "February" etc. if user enters invalid month number your code should print "Invalid number"

```
int month = 8;
String monthString;
switch (month) {
        case 1:  monthString = "January";
              break;
        case 2:  monthString = "February";
              break;
        case 3:  monthString = "March";
              break;
        case 4:  monthString = "April";
              break;
        case 5:  monthString = "May";
              break;
        case 6:  monthString = "June";
              break;
        case 7:  monthString = "July";
              break;
        case 8:  monthString = "August";
              break;
        case 9:  monthString = "September";
              break;
        case 10: monthString = "October";
              break;
        case 11: monthString = "November";
              break;
        case 12: monthString = "December";
              break;
        default: monthString = "Invalid month";
              break;
}
System.out.println(monthString);
```

3) Type a code which prints the all month names starting with the given month numbers. Example: If the month number is 9 output will be September October November December

```
List<String> futureMonths = new ArrayList<>();
int month = 8;
switch (month) {
  case 1:  futureMonths.add("January");
  case 2:  futureMonths.add("February");
  case 3:  futureMonths.add("March");
  case 4:  futureMonths.add("April");
  case 5:  futureMonths.add("May");
  case 6:  futureMonths.add("June");
  case 7:  futureMonths.add("July");
  case 8:  futureMonths.add("August");
  case 9:  futureMonths.add("September");
  case 10: futureMonths.add("October");
  case 11: futureMonths.add("November");
  case 12: futureMonths.add("December");
           break;
  default: break;
}

if (futureMonths.isEmpty()) {
  System.out.println("Invalid month number");
} else {
  for (String monthName : futureMonths) {
    System.out.print(monthName + " ");
  }
}
```

4) **Print "Boy" if the gender is "Male" (Ignore cases)**
Print "Girl" if the gender is "Female" (Ignore cases)
Print "Others" if the gender is different from "Male" and "Female"

```
String gender = "Male";
switch(gender.toLowerCase()){
  case "female":
     System.out.println("Girl");
     break;
  case "male":
     System.out.println("Boy");
     break;
  default:
     System.out.println("Others");
}
```

5) **Type a code to display the number of days in a given month of a given year.**
Example: The number of days in February of 2000 was 29.

```
int month = 2;
int year = 2000;
int numDays = 0;

switch (month) {
  case 1: case 3: case 5:
  case 7: case 8: case 10:
  case 12:
    numDays = 31;
    break;
  case 4: case 6:
  case 9: case 11:
    numDays = 30;
    break;
  case 2:
    if (((year % 4 == 0) && !(year % 100 == 0)) || (year % 400 == 0)) {
      numDays = 29;
    }else {
      numDays = 28;
    }
    break;
  default:
    System.out.println("Invalid month.");
    break;
}
System.out.println("Number of Days = " + numDays);
```

6) **Use switch statement to create a simple calculator which does addition, subtraction, multiplication, and division with any 2 numbers**

a) When user entered 10.2 and 5 and + sign your code should print "The result is 15.2" on the console

b) When user entered 10 and 5 and - sign your code should print "The result is 5.0" on the console

c) When user entered 10 and 5.3 and * sign your code should print "The result is 53.0" on the console

d) When user entered 10 and -5 and / sign your code should print "The result is -2.0" on the console

e) When user entered 10 and -5 and operation different from +, -, *, / your code should print "That operation cannot be done"

```
double a = 10.2;
double b = 5;
String operator = "+";
switch(operator){
  case "+":
     System.out.println(a+b);
     break;
  case "-":
     System.out.println(a-b);
     break;
  case "*":
     System.out.println(a*b);
     break;
  case "/":
     System.out.println(a/b);
     break;
  default:
     System.out.println("Invalid operator");
}
```

7) Use switch statement to create a converter which converts mile to km, second to hour, fahrenheit to celsius. Find the formulas for conversions from Google.

a) When user entered 10 and mileToKm String, your code should print "16 km" (The number will be dynamic) on the console

b) When user entered 7200 and secondsToHours String, your code should print "2 Hours"(The number will be dynamic) on the console

c) When user entered 83 and fahrenheitToCelsius String, your code should print "28.3333 celsius"(The number will be dynamic) on the console

d) When user entered operation different from mileToKm, secondsToHours, fahrenheitToCelsius your code should print "That operation was not defined for that converter"

```
double mile = 10;
double seconds = 7200;
double fahrenheit = 83;
String operator = "fahrenheitToCelsius";
switch(operator){
  case "mileToKm":
    System.out.println(mile*1.60934);
    break;
  case "secondsToHours":
    System.out.println((seconds/60)/60);
    break;
  case "fahrenheitToCelsius":
    System.out.println((fahrenheit-32)*5/9);
    break;
  default:
    System.out.println("That operation was not defined for that converter");
}
```

8) **Create an Enum and put the browser names in it like CHROME, SAFARI, IE, FIREFOX, YANDEX. Use switch statement and print 'I am using CHROME' for CHROME, 'I am using SAFARI' for SAFARI, 'I am using IE for IE', 'I am using FIREFOX for FIREFOX', and 'I am using YANDEX' for YANDEX, 'Not valid browser' for browsers different from CHROME, SAFARI, IE, FIREFOX, YANDEX.**

Note: You should create enum on developer console

```
public enum Browsers{
    CHROME, SAFARI, IE, FIREFOX, YANDEX
}


String browser = Browsers.CHROME.toString();
switch(browser){
  case "CHROME":
    System.out.println("I am using CHROME");
    break;
  case "SAFARI":
    System.out.println("I am using SAFARI");
    break;
  case "IE":
    System.out.println("I am using IE");
    break;
  case "FIREFOX":
    System.out.println("I am using FIREFOX");
    break;
  case "YANDEX":
    System.out.println("I am using YANDEX");
    break;
  default:
    System.out.println("It is not valid browser");
}
```

9) Create an Enum whose name is 'Seasons' and print the following by using the Enum in switch-on statement

For winter print 'Snowboarding'

For summer and spring print 'Fishing'

For fall print 'Trekking'

```
public enum Seasons {
   WINTER, SUMMER, FALL, SPRING
}

String browser = Seasons.WINTER.toString();
switch(browser){
  case "WINTER":
     System.out.println("Snowboarding");
     break;
  case "SUMMER":
  case "SPRING":
     System.out.println("Fishing");
     break;
  case "FALL":
     System.out.println("Trekking");
     break;
  default:
     System.out.println("Undefined season");
}
```

10) Create a method which gives you a random alphabetical character in myClass Class then print the following by using the method.

Print "First Character" for 'A' and 'a'

Print "Second Character" for 'B' and 'b'

Print "Third Character" for 'C' and 'c'

Print "Forth Character" for 'D' and 'd'

Print "Other Characters" for all others

```
public class myClass {
  public static String getRandomAlphabet(){
    String alphabets = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz";
    int maxIndex = alphabets.length()-1;
    int randomIndex = (int)Math.round(Math.random()*maxIndex);
    return alphabets.substring(randomIndex, randomIndex+1);
  }
}

String alphabet = myClass.getRandomAlphabet().toLowerCase();
switch(alphabet){
  case "a":
    System.out.println("First Character");
    break;
  case "b":
    System.out.println("Second Character");
    break;
  case "c":
    System.out.println("Third Character");
    break;
  case "d":
    System.out.println("Fourth Character");
    break;
  default:
    System.out.println("Other Characters");
}
```

1) **Type all integers which are divisible by 4 and divisible by 6 from 120 to 11 in the same line with a space between two consecutive integers**

1.Way:

```
String s = "";
for(Integer i=120; i>10; i--){
  if(i%4==0 && i%6==0){
     s = s + i + " ";
  }
}
System.out.println(s);
```

2.Way:

```
String s = "";
Integer i=120;
while(i>10){
  if(i%4==0 && i%6==0){
     s = s + i + " ";
  }
  i--;
}
System.out.println(s);
```

3.Way:

```
String s = "";
Integer i=120;
do{
  if(i%4==0 && i%6==0){
     s = s + i + " ";
  }
  i--;
}while(i>10);
System.out.println(s);
```

2) **Type code to print repeated characters in a String. For example; accessories   ces**

1.Way:

```
String s = "accessories";
String d = "";
for(Integer i=0; i<s.length(); i++){
  String c = s.substring(i,i+1);
  if(s.indexOf(c)!=s.lastIndexOf(c)){
    if(!d.contains(c)){
      d = d + c;
    }
  }
}
System.out.println(d);
```

2.Way:

```
String s = "accessories";
String d = "";
Integer i=0;
while(i<s.length()){
  String c = s.substring(i,i+1);
  if(s.indexOf(c)!=s.lastIndexOf(c)){
    if(!d.contains(c)){
      d = d + c;
    }
  }
  i++;
}
System.out.println(d);
```

3.Way:

```
String s = "accessories";
String d = "";
Integer i=0;
do{
  String c = s.substring(i,i+1);
  if(s.indexOf(c)!=s.lastIndexOf(c)){
    if(!d.contains(c)){
      d = d + c;
    }
  }
  i++;
}while(i<s.length());
System.out.println(d);
```

3) **Type code to check whether a String is palindrome or not. If a String is the same with its reverse then it is called palindrome. For example;** “anna”, “123321” **are palindromes**

1.Way:

```
String s = "runnur";
String r = "";
for(Integer i=s.length()-1; i>-1; i--){
  String c = s.substring(i,i+1);
  r = r + c;
}
if(s.equals(r)){
  System.out.println(s + " is palindrome");
}else{
  System.out.println(s + " is not palindrome");
}
```

2.Way:

```
String s = "runnur";
String r = "";
Integer i=s.length()-1;
while(i>-1){
  String c = s.substring(i,i+1);
  r = r + c;
  i--;
}
if(s.equals(r)){
  System.out.println(s + " is palindrome");
}else{
  System.out.println(s + " is not palindrome");
}
```

3.Way:

```
String s = "runnur";
String r = "";
Integer i=s.length()-1;
do{
  String c = s.substring(i,i+1);
  r = r + c;
  i--;
}while(i>-1);
if(s.equals(r)){
  System.out.println(s + " is palindrome");
}else{
  System.out.println(s + " is not palindrome");
}
```

4) Type code to print the unique digits in an integer. Example; 223878   37

1.Way:

```
Integer num = 223878;
String s = String.valueOf(num);
String d = "";
for(Integer i=0; i<s.length(); i++){
  String c = s.substring(i,i+1);
  if(s.indexOf(c)==s.lastIndexOf(c)){
     d = d + c;
  }
}
System.out.println(d);
```

2.Way:

```
Integer num = 223878;
String s = String.valueOf(num);
String d = "";
Integer i = 0;
while(i<s.length()){
  String c = s.substring(i,i+1);
  if(s.indexOf(c)==s.lastIndexOf(c)){
     d = d + c;
  }
  i++;
}
System.out.println(d);
```

3.Way:

```
Integer num = 223878;
String s = String.valueOf(num);
String d = "";
Integer i = 0;
do{
  String c = s.substring(i,i+1);
  if(s.indexOf(c)==s.lastIndexOf(c)){
    d = d + c;
  }
  i++;
}while(i<s.length());
System.out.println(d);
```

5) Type code to draw the following image by using a for loop.

```
A A A A A
A A A A A
A A A A A
```

1.Way:

```
int rows = 3;
int columns = 5;
for(int i=1; i<=rows; i++){
  String s = "";
  for(Integer k=1; k<=columns; k++){
    s = s + "A ";
  }
  System.out.println(s);
}
```

2.Way:

```
int rows = 3;
int columns = 5;
int i=1;
while(i<=rows){
  String s = "";
  for(Integer k=1; k<=columns; k++){
     s = s + "A ";
  }
  System.out.println(s);
  i++;
}
```

3.Way:

```
int rows = 3;
int columns = 5;
int i=1;
do{
  String s = "";
  for(Integer k=1; k<=columns; k++){
     s = s + "A ";
  }
  System.out.println(s);
  i++;
}while(i<=rows);
```

6) **Type code to draw the following image by using a for loop.**

```
A
A A
A A A
A A A A
```

1.Way:

```
int rows = 4;
for(Integer i=1; i<=rows; i++){
  String s = "";
  for(Integer k=1; k<=i; k++){
     s = s + "A ";
  }
  System.out.println(s);
}
```

2.Way:

```
int rows = 4;
int i=1;
while(i<=rows){
  String s = "";
  for(int k=1; k<=i; k++){
     s = s + "A ";
  }
  System.out.println(s);
  i++;
}
```

3.Way:

```
int rows = 4;
int i=1;
do{
  String s = "";
  for(int k=1; k<=i; k++){
     s = s + "A ";
  }
  System.out.println(s);
  i++;
}while(i<=rows);
```

7) **Type code to create multiplication table like**
3x1=3 3x2=6 3x3=9 3x4=12 3x5=15 3x6=18 3x7=21 3x8=24 3x9=27 3x10=30

1.Way:

```
int num = 3;
for(int i=1; i<11; i++){
  System.out.println(num + "x" + i + "=" + num*i);
}
```

2.Way:

```
int num = 3;
int i=1;
while(i<11){
  System.out.println(num + "x" + i + "=" + num*i);
  i++;
}
```

3.Way:

```
int num = 3;
int i=1;
do{
  System.out.println(num + "x" + i + "=" + num*i);
  i++;
}while(i<11);
```

8) **Type code to print odd integers from 20 to 3 in the same line with a space between two consecutive ones.**

1.Way:

```
String s = "";
for(Integer i=20; i>2; i--){
  if(i%2!=0){
     s = s + i + " ";
  }
}
System.out.println(s);
```

2.Way:

```
String s = "";
int i=20;
while(i>2) {
  if(i%2!=0){
     s = s + i + " ";
  }
  i--;
}
System.out.println(s);
```

3.Way:

```
String s = "";
int i=20;
do{
  if(i%2!=0){
     s = s + i + " ";
  }
  i--;
}while(i>2);
System.out.println(s);
```

9) **Type code to print all lowercase characters in a String with an asterix.**
For example; 'Ali Can?' ==> l*i*a*n*

1.Way:

```
String s = "Ali Can?";
s = s.replaceAll("[⊥a-z]", "");
String t = "";
for(Integer i=0; i<s.length(); i++){
  String r = s.substring(i,i+1);
  t = t + r + '*';
}
System.out.println(t);
```

2.Way:

```
String s = "Ali Can?";
s = s.replaceAll("[⊥a-z]", "");
String t = "";
Integer i=0;
while(i<s.length()){
  String r = s.substring(i,i+1);
  t = t + r + '*';
  i++;
}
System.out.println(t);
```

3.Way:

```
String s = "Ali Can?";
s = s.replaceAll("[⊥a-z]", "");
String t = "";
Integer i=0;
do{
  String r = s.substring(i,i+1);
  t = t + r + '*';
  i++;
}while(i<s.length());
System.out.println(t);
```

10) **Type code to print digits just in the decimal part of the given decimal number with an asterix. For example; 75.4238 ⇒ *4*2*3*8**

1.Way:

```
double num = 75.4238;
String s = String.valueOf(num);
int idxOfComma = s.indexOf(".");
String decimalPart = s.substring(idxOfComma + 1);
String t = "";
for(Integer i=0; i<decimalPart.length(); i++){
  String r = decimalPart.substring(i,i+1);
  t = t + "*" + r;
}
System.out.println(t);
```

2.Way:

```
double num = 75.4238;
String s = String.valueOf(num);
int idxOfComma = s.indexOf(".");
String decimalPart = s.substring(idxOfComma + 1);
String t = "";
Integer i=0;
while(i<decimalPart.length()){
  String r = decimalPart.substring(i,i+1);
  t = t + '*' + r;
  i++;
}
System.out.println(t);
```

3.Way:

```
double num = 75.4238;
String s = String.valueOf(num);
int idxOfComma = s.indexOf(".");
String decimalPart = s.substring(idxOfComma + 1);
String t = "";
Integer i=0;
do{
  String r = decimalPart.substring(i,i+1);
  t = t + '*' + r;
  i++;
}while(i<decimalPart.length());
System.out.println(t);
```

11) Type code to reverse a String. Example; Mark ==> kraM

1.Way:

```
String t = "Mark";
String rev = "";
for(Integer i=t.length()-1; i>-1; i--){
  rev = rev + t.substring(i,i+1);
}
System.out.println(rev);
```

2.Way:

```
String t = "Mark";
StringBuilder sb = new StringBuilder(t);
System.out.println(sb.reverse());
```

12) Type code to draw the following image by using a for loop.

```
A A A A A A A A
A X X X X X X A
A X X X X X X A
A X X X X X X A
```

1.Way:

```
int rows = 4, columns = 8;
for(int i=1; i<=rows; i++){
  String s = "";
  if(i==1 || i==rows){
    for(Integer k=1; k<=columns; k++){
      s = s + "A ";
    }
    System.out.println(s);
  }else{
    s = s + "A ";
    for(Integer m=2; m<columns; m++){
      s = s + "X ";
    }
    s = s + "A ";
    System.out.println(s);
  }
}
```

2.Way:

```
int rows = 4, columns = 8, i = 1;

while(i<=rows){
  String s = "";
  if(i==1 || i==rows){
     for(Integer k=1; k<=columns; k++){
        s = s + "A ";
     }
     System.out.println(s);
  }else{
     s = s + "A ";
     for(Integer m=2; m<columns; m++){
        s = s + "X ";
     }
     s = s + "A ";
     System.out.println(s);
  }
  i++;
}
```

3.Way:

```
int rows = 4, columns = 8, i = 1;

do{
  String s = "";
  if(i==1 || i==rows){
    for(Integer k=1; k<=columns; k++){
      s = s + "A ";
    }
    System.out.println(s);
  }else{
    s = s + "A ";
    for(Integer m=2; m<columns; m++){
      s = s + "X ";
    }
    s = s + "A ";
    System.out.println(s);
  }
  i++;
}while(i<=rows);
```

13) Type code to find the sum of the integers from 3 to 14

1.Way:

```
int sum = 0;
for(int i=3; i<15; i++){
  sum = sum+i;
}
System.out.println(sum);
```

2.Way:

```
int sum = 0, i = 3;
while(i<15){
  sum = sum+i;
  i++;
}
System.out.println(sum);
```

3.Way:

```
int sum = 0, i = 3;
do{
  sum = sum+i;
  i++;
}while(i<15);
System.out.println(sum);
```

14) Type code to find the multiplication of the integers from 3 to 9

1.Way:

```
int m = 1;
for(int i=3; i<10; i++){
  m = m*i;
}
System.out.println(m);
```

2.Way:

```
int m = 1, i = 3;
while(i<10){
  m = m*i;
  i++;
}
System.out.println(m);
```

3.Way:

```
int m = 1, i = 3;
do{
  m = m*i;
  i++;
}while(i<10);
System.out.println(m);
```

15) Type code to print characters from 'C' to 'A' on the console by using do-while loop

1.Way:

```
String result = "";
for(char i = 'C'; i>='A'; i--){
  result = result + i;
}
System.out.println(result);
```

2.Way:

```
char ch = 'C';
String result = "";
while(ch>='A'){
  result = result + ch;
  ch--;
}
System.out.println(result);
```

3.Way:

```
char ch = 'C';
String result = "";
do{
  result = result + ch;
  ch--;
}while(ch>='A');
System.out.println(result);
```

16) Type all characters before the first occurrence of 'm' in a String

1.Way:

```
String s = "Christmas";
String result = "";
for(int i=0; i<s.length(); i++){
  String c = s.substring(i,i+1);
  if(c.equals("m")){
     break;
  }
  result = result + c;
}
System.out.println(result);
```

2.Way:

```
String s = "Christmas";
String result = "";
int i=0;
while( i<s.length()){
  String c = s.substring(i,i+1);
  if(c.equals("m")){
     break;
  }
  result = result + c;;
  i++;
}
System.out.println(result);
```

3.Way:

```
String s = "Christmas";
String result = "";
int i=0;
do{
  String c = s.substring(i,i+1);
  if(c.equals("m")){
     break;
  }
  result = result + c;;
  i++;
}while( i<s.length());
System.out.println(result);
```

17) Type code to find the sum of the digits in an integer

1.Way:

```
int n = 753;
int sumOfDigits = 0;

for(int i = n; i>0; i=i/10){
  sumOfDigits = sumOfDigits + i%10;
}
System.out.println(sumOfDigits);
```

2.Way:

```
int num = 753;
int sumOfDigits = 0;
while(num>0){
  sumOfDigits = sumOfDigits + num%10;
  num = num/10;
}
System.out.println(sumOfDigits);
```

3.Way:

```
int num = 753;
int sumOfDigits = 0;
do{
  sumOfDigits = sumOfDigits + num%10;
  num = num/10;
}while(num>0);
System.out.println(sumOfDigits);
```

18) Type code to print unique characters in a String. For example; Hello ==> Heo

1.Way:

```
String s = "Hello";
for(int i=0; i<s.length();i++){
  String c = s.substring(i,i+1);
  if(s.indexOf(c) == s.lastIndexOf(c)){
     System.out.print(c);
  }
}
```

2.Way:

```
String s = "Hello";
int i = 0;
while(i<s.length()){
  String c = s.substring(i,i+1);
  if(s.indexOf(c) == s.lastIndexOf(c)){
     System.out.print(c);
  }
  i++;
}
```

3.Way:

```
String s = "Hello";
int i = 0;
do{
  String c = s.substring(i,i+1);
  if(s.indexOf(c) == s.lastIndexOf(c)){
     System.out.print(c);
  }
  i++;
}while(i<s.length());
```

19) Find the total number of characters different from space and punctuation marks in a String

```
String s = "Java is a strongly typed, object-oriented programming language.";
s = s.replaceAll("\\s","").replaceAll("\\p{Punct}", "");
System.out.println(s.length());
```

20) Type code to draw the following image by using a for loop.

```
* * * * * *
* * * * *
* * * *
* * *
* *
*
```

```
int term=6;
String s = "";

for(int i=1; i<=term; i++){
  for(int k=term; k>=i; k--){
     s = s + "* ";
  }
  System.out.println(s);
  s = "";
}
```

21) Type code to find the sum of the unique digits of an integer
Example: 12133455 ⇒ 2+4=6

```
int p = 12133455;
String r = String.valueOf(p);
int sumOfUniqueDigits = 0;

for(int i=0; i<r.length(); i++){
  String c = r.substring(i,i+1);
  if(r.indexOf(c)==r.lastIndexOf(c)){
     sumOfUniqueDigits = sumOfUniqueDigits + Integer.valueOf(c);
  }
}
System.out.println(sumOfUniqueDigits);
```

22) Type code to print integers from 3 to 9 except 5

```
for(Integer i=3; i<10; i++){
  if(i==5){
     continue;
  }
  System.out.print(i + " ");
}
```

23) Ask the user to enter an integer to check if it is a prime number or not. Example: if user enters 43 output will be “43 is a prime number”**, if user enters 120 output will be** “120 is not a prime number”**, if user enters negative integers output will be** “Enter a positive integer”

```
Scanner scan = new Scanner(System.in);
System.out.println("Enter a positive integer to check if it is prime or not");
int number = scan.nextInt();


int count = 0;
if(number>0) {
  if (number == 1) {
    System.out.println(number + " is prime number");
  }else {
    for(int i=2; i<number; i++) {
      if(number%i==0) {
        count++;
      }
    }
    if(count==0) {
      System.out.println(number + " is prime");
    }else {
      System.out.println(number + " is not prime");
    }
  }
}else {
  System.out.println("Enter a positive integer");
}
```

24) Ask the user how many fibonacci numbers he wants to see. Then write the code that prints these fibonacci numbers. Example: if user enters 6 output will be 1 1 2 3 5 8

```
Scanner scan = new Scanner(System.in);
System.out.println("How many fibonacci number do you want to see?");
int number = scan.nextInt();

int fibo1 = 1;
int fibo2 = 1;
int fibonacci = 0;
System.out.print(fibo1 + " ");
System.out.print(fibo2 + " ");

for (int i = 1; i < number-1; i++) {
  fibonacci = fibo1 + fibo2;
  fibo1 = fibo2;
  fibo2 = fibonacci;
  System.out.print(fibonacci + " ");
}
```

25) A number is called an Armstrong number if it equals the sum of the cube of its every digit. For example, 153 is an Armstrong number because of 153= 1 + 125 + 27, which equals the cube of every digit. Type a code to check if the given number is the Armstrong number or not.

```
Scanner scan = new Scanner(System.in);
System.out.println("Enter an integer to check if it is Armstrong number or not");
int n = scan.nextInt();

int sumOfCubes = 0;
int temp = n;
int digit = 0;

while(n>0) {
  digit = n%10;
  sumOfCubes = sumOfCubes + digit*digit*digit;
  n = n/10;
}

if(temp==sumOfCubes) {
  System.out.println(temp + " is Armstrong number");
}else {
  System.out.println(temp + " is not Armstrong number");
}
```

26) **Get a String and a character from user**

Count the number of characters between the first occurrence and the last occurrence of the given character in the String

Do not count the space characters

If the character which user selected is displayed just once in the String return -1

If the character which user selected does not exist in the String return -1

For example; "Java is easy" - 'a' ==> 5

"Java is easy" - 'w' ==> -1

"Java is easy" - 'e' ==> -1

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

System.out.println("Enter a String...");
String s = scan.nextLine();

System.out.println("Enter a character...");
char ch = scan.next().charAt(0);

int firstOccIdx = s.indexOf(ch);
int lastOccIdx = s.lastIndexOf(ch);
int counter = 0;

if(firstOccIdx == lastOccIdx) {
  System.out.println(-1);
}else {
  for(int i=firstOccIdx+1; i<lastOccIdx; i++) {
    if(s.charAt(i)!=' '){
      counter++;
    }
  }
  System.out.println(counter);
}
```

27) From a given array find all pairs whose sum is a given number.
If the array is {4, 6, 5, -10, 8, 5, 20} and the number is 10 then output will be

4+6=10, 5+5=10, -10+20=10

```
Integer arr[] = {4, 6, 5, -10, 8, 5, 20};
int num = 10;

for(int i=0; i<arr.length; i++) {
  for(int j=i+1; j<arr.length; j++) {
    if(arr[i]+arr[j]==num) {
      System.out.println(arr[i] + " + " + arr[j] + " = " + num );
    }
  }
}
```

28) Type a code that prints all prime numbers less than a given number. Example: if user enters 20 output will be 2, 3, 5, 7, 11, 13, 17, 19

```
public static void main(String[] args) {
  Scanner scan = new Scanner(System.in);
  System.out.println("Enter a positive integer");
  int n = scan.nextInt();

  printPrime(n);
}
private static boolean isPrime(int n){
  if (n <= 1) {
     return false;
  }
  for (int i = 2; i < n; i++) {
     if (n % i == 0) {
        return false;
     }
  }
  return true;
}
public static void printPrime(int n){
  for (int i = 2; i <= n; i++){
     if(isPrime(i)) {
        System.out.print(i + " ");
     }
  }
}
```

29) Type a program to print the integers from 1 to 100 without using any loop.

```
public static void main(String[] args) {
  printNumbers(100);
}

static void printNumbers(int n) {
  if(n > 0) {
    printNumbers(n - 1);
    System.out.print(n + " ");
  }
  return;
}
```

30) Each line of input contains one or several words separated with single spaces. Type code to reverse the order of words in each line of input, while preserving the words themselves. The lines of your output should not have any trailing or leading spaces.

Example; Kemal Can Kuzu will be converted to Kuzu Can Kemal

```
public static void main(String[] args) {
  String input = "Ali\n" +
                 "Veli Han\n" +
                 "Kemal Can Kuzu";
  String rev = "";
  String[] lines = input.split(System.getProperty("line.separator"));
  for(String w : lines) {
    String[] words = w.split(" ");
    for (int i = words.length - 1; i >= 0; i--) {
      if (i != 0) {
        rev = rev + words[i] + " ";
      } else {
        rev = rev + words[i];
      }
    }
    System.out.println(rev);
    rev="";
  }
}
```

31) Each line of input contains one or several words separated with single spaces. Type code to reverse the order of words except the first and the last words in each line of input, while preserving the words themselves. The lines of your output should not have any trailing or leading spaces. Example; Kemal Can Tan Han Kuzu will be converted to Kemal Han Tan Can Kuzu

```
String input = "Ali\n" +
    "Veli Can Han\n" +
    "Kemal Can Tan Kuzu\n" +
    "Kemal Can Tan Han Kuzu\n" +
    "Kemal Can Tan Han Man Kuzu";

String[] lines = input.split(System.getProperty("line.separator"));
for(String w : lines) {
  String[] words = w.split(" ");
  String rev = words[0] + " ";
  for (int i = words.length - 2; i >= 1; i--) {
       rev = rev + words[i] + " ";
  }
  rev = rev + words[words.length-1];
  System.out.println(rev);
  rev="";
}
```

1) Find the middle element in an integer array

Example: (12, 5, 8) ==> (5, 8, 12) ==> Output=8

(12, 5, 8, 13) ==> (5, 8, 12, 13) ==> Output=(8+12)/2 = 10

```
int[ ] a = new int[ ]{12, 5, 8, 13};
Arrays.sort(a);

if(a.length%2!=0){
  Integer indexOfMiddleElement = a.length/2;
  System.out.println(a[indexOfMiddleElement]);
}else{
  Integer indexOfMiddleElement = a.length/2;
  Integer averageValue = (a[indexOfMiddleElement] + a[indexOfMiddleElement-1])/2;
  System.out.println(averageValue);
}
```

2) **Find the smallest positive element and greatest negative element in an integer array Example:(-12, 18, -5, 23, -2) ==> Smallest positive is 18, greatest negative is -2**

1.Way:

```
int[] a = new int[]{-12, 18, -5, 23, -2};
Arrays.sort(a);

Integer minPositive = a[a.length-1];
Integer maxNegative = a[0];

for(Integer w : a){
  if(w>=0){
     minPositive = Math.min(minPositive, w);
  }

  if(w<0){
     maxNegative = Math.max(maxNegative, w);
  }
}
System.out.println("Minimum positive: " + minPositive);
System.out.println("Maximum negative: " + maxNegative);
```

2.Way:

```
int[] a = new int[]{-12, 18, -5, 23, -2};
Arrays.sort(a);
for (Integer i=0;i<a.length;i++){
  if (a[i]<0 && a[i+1]>0) {
     System.out.println("Maximum negative: " + a[i]);
     System.out.println("Minimum positive: " + a[i+1]);
  }
}
```

3) Find the elements whose length is the smallest in a String array
Example: (Kemal, Jonathan, Mark, Angie, Veli) ==> Output is Mark, Veli

```
String[] b = new String[]{"Kemal", "Jonathan", "Mark", "Angie", "Veli"};
int minLength = b[0].length();

for(String w : b){
  minLength = Math.min(minLength, w.length());
}
for(String w : b){
  if(minLength == w.length()){
    System.out.println(w);
  }
}
```

4) Get the initials of the elements in a String array, if the String ends with 'n' or 'k'
Example: { “Kemal”, “Jonathan”, “Mark”, “Jackson”, “Ali” } ==> Output is JJM

```
String[] b = new String[]{"Kemal", "Jonathan", "Mark", "Angie", "Veli"};
String initials = "";

for(String w : b){
  if(w.endsWith("n") || w.endsWith("k")){
    initials = initials + w.substring(0,1);
  }
}
System.out.println(initials);
```

5) Find the total number of characters used in String array elements
Example: { "Kemal", "Jonathan", "Mark", "Angie", "Veli" } ==> Output is 26

```
String[] b = new String[]{"Kemal", "Jonathan", "Mark", "Angie", "Veli"};
int sum = 0;

for(String w : b){
  sum = sum + w.length();
}
System.out.println("Total number of characters is " + sum);
```

6) Count how many words start with 'a' or 'A' in a given String

```
String s = "Apex is an object oriented programming language";
String[] arr = s.split(" ");
int counter = 0;
for(String w : arr){
  if(w.startsWith("a") || w.startsWith("A")){
     counter++;
  }
}
System.out.println("The number of words which starts with a or A is : " + counter);
```

7) Find the number of vowels in a String

```
String s = "Apex is an object oriented programming language";
String[] arr = s.toLowerCase().split("");
int counter = 0;
for(String w : arr){
  switch(w){
     case "a":
     case "e":
     case "i":
     case "o":
     case "u":
        counter++;

  }
}
System.out.println("The number of vowels is " + counter);
```

8) Type code to find array elements whose first and last characters are same

```
String[] arr = new String[]{"alabama", "pick", "sos", "sets", "pex"};
int size = arr.length;
for(Integer i=0; i<size; i++){
  String firstChar = arr[i].substring(0, 1);
  String lastChar = arr[i].substring(arr[i].length()-1);
  if(firstChar.equals(lastChar)){
     System.out.println(arr[i]);
  }
}
```

9) **Type code to find if a given element exists in a given array or not**

```
String[] arr = new String[]{"Apex", "is", "an", "object", "oriented", "programming", "language"};
String s = "object";
int counter = 0;
for(String w : arr){
  if(w.equalsIgnoreCase(s)){
    counter++;
  }
}
if(counter>0){
  System.out.println(s + " exists");
}else{
  System.out.println(s + " does not exist");
}
```

10) **Type code to find sum of the number of characters of array elements**

```
String[] arr = new String[]{"alabama", "pick", "sos", "sets", "pex"};
Integer sum = 0;
for(String w : arr){
  sum = sum + w.length();
}
System.out.println(sum);
```

11) Type code to put all zeros to end in an integer array.
Example: (5, 0, 2, 0, 3) ==> (5, 2, 3, 0, 0)

```
Integer[] arr = new Integer[]{5, 0, 2, 0, 3};
Integer[] brr = new Integer[arr.length];
int firstIdx = 0;
int lastIdx = arr.length-1;

for(int i=0; i<arr.length; i++){
  if(arr[i]!=0){
     brr[firstIdx] = arr[i];
     firstIdx++;
  }else{
     brr[lastIdx]=0;
     lastIdx--;
  }
}
System.out.println(Arrays.toString(brr));
```

12) Create an integer array by getting the elements from the user then print the difference between the smallest and the greatest elements on the console.

```
Scanner scan = new Scanner(System.in);
System.out.println("Enter the length of the array");
int len = scan.nextInt();

int arr[] = new int[len];

System.out.println("Enter " + len + " array elements");
for(int i=0; i<len; i++) {
  arr[i] = scan.nextInt();
}
System.out.println(Arrays.toString(arr));

Arrays.sort(arr);
System.out.println(Arrays.toString(arr));

int diffOfBigAndSmall = arr[arr.length-1] - arr[0];
System.out.println("Difference between the greatest and the lowest is " + diffOfBigAndSmall);
```

13) Ask the user to enter 2 Strings. If the characters and the numbers of characters of the Strings are the same then print "Anagram" on the console. Otherwise, print "Not Anagram" on the console. Ignore cases. For example; "Mary" and "army" and "RAMY" are Anagrams.

```
Scanner scan = new Scanner(System.in);
System.out.println("Enter two Strings");
String s1 = scan.nextLine();
String s2 = scan.nextLine();

String a1[] = s1.toLowerCase().split("");
Arrays.sort(a1);

String a2[] = s2.toLowerCase().split("");
Arrays.sort(a2);

if (s1.length() != s2.length()) {
  System.out.println("Not anagram");
} else if (s1.isEmpty() || s2.isEmpty()) {
  System.out.println("Not anagram");
} else if (Arrays.equals(a1, a2)) {
  System.out.println("Anagram");
} else {
  System.out.println("Not anagram");
}
```

1) Find the sum of all list elements in an integer list

```
List<Integer> myList = new ArrayList<>();
myList.add(12);
myList.add(31);
myList.add(7);
myList.add(13);
myList.add(10);
int sum = 0;
for(Integer w : myList){
  sum = sum + w;
}
System.out.println(sum);
```

2) Find the sum of all list elements before the first occurrence of 13 in an integer list

```
List<Integer> myList = new ArrayList<>();
myList.add(12);
myList.add(31);
myList.add(7);
myList.add(13);
myList.add(10);
int sum = 0;
for(Integer w : myList){
  if(w==13){
     break;
  }
  sum = sum + w;
}
System.out.println(sum);
```

3) Find the multiplication of all even list elements in an integer list

1.Way:

```
List<Integer> myList = new ArrayList<>();
myList.add(12);
myList.add(31);
myList.add(7);
myList.add(13);
myList.add(10);
int m = 1;
for(Integer w : myList){
  if(w%2!=0){
     continue;
  }
  m = m * w;
}
System.out.println(m);
```

2.Way:

```
List<Integer> myList = new ArrayList<>();
myList.add(12);
myList.add(31);
myList.add(7);
myList.add(13);
myList.add(10);
int m = 1;
for(Integer w : myList){
  if(w%2==0){
     m = m * w;
  }
}
System.out.println(m);
```

4) Check if elements are in descending order in a list

Example: (Yellow, Blue, Red, Green) ==> Output: It is not in descending order

(Yellow, Red, Green, Blue) ==> Output: It is in descending order

```
List<String> e = new ArrayList<>();
e.add("Yellow");
e.add("Red");
e.add("Green");
e.add("Blue");

List<String> f = new ArrayList<>();
f.addAll(e);
Collections.sort(f);
System.out.println(e);
System.out.println(f);

int size = e.size();
int flag = 0;

for(int i=0; i<size; i++){
  if(f.get(i).equals(e.get(size-1-i))){
     flag++;
  }
}
if(flag == size){
  System.out.println("It is in descending order");
}else{
  System.out.println("It is not in descending order");
}
```

5) If the list has 15 as element, change all 15s to 51
Example: (12, 11, 15, 34, 43) ==> Output is (12, 11, 51, 34, 43)

```
List<Integer> g = new ArrayList<>();
g.add(12);
g.add(11);
g.add(15);
g.add(34);
g.add(15);
g.add(43);

if(g.contains(15)){
  for(int w : g){
    if(w==15){
      int idx = g.indexOf(15);
      g.set(idx, 51);
    }
  }
  System.out.println(g);
}else{
  System.out.println("There is no expected element");
}
```

6) **If a list has 15 or 13, remove them.**
Example: (10, 31, 15, 13, 54) ==> Output is (10, 31, 54)

```
List<Integer> h = new ArrayList<>();
h.add(10);
h.add(31);
h.add(15);
h.add(13);
h.add(54);
h.add(13);

if(!h.contains(15) && !h.contains(13)){
  System.out.println("The list does not contain 13 and 15");
}else {
  for (int i = 0; i < h.size(); i++) {
    if (h.get(i) == 15) {
      int idx15 = h.indexOf(15);
      h.remove(idx15);
      i--;
    }
    if (h.get(i) == 13) {
      int idx13 = h.indexOf(13);
      h.remove(idx13);
      i--;
    }
  }
  System.out.println(h);
}
```

7) Find the closest 2 integers in an Integer List
Example: (12, 31, 15, 13, 54) ==> Output is 12 and 13

```
List<Integer> h = new ArrayList<>();
h.add(12);
h.add(31);
h.add(15);
h.add(13);
h.add(54);

Collections.sort(h);
int minDiff = h.get(2) - h.get(1);

for(int i=1; i<h.size(); i++){
  minDiff = Math.min(minDiff, h.get(i)-h.get(i-1));
}
for(Integer i=1; i<h.size(); i++){
  if(h.get(i)-h.get(i-1) == minDiff){
     System.out.println(h.get(i) + " and " + h.get(i-1));
  }
}
```

8) Increase the value of every element except 7 and 10 by 2 in a List
Example: (12, 31, 7, 13, 10) ==> Output is (14, 33, 7, 15, 10)

```
List<Integer> h = new ArrayList<>();
h.add(12);
h.add(31);
h.add(7);
h.add(13);
h.add(10);

for(Integer w : h){
  if(w==7 || w==10){
     continue;
  }
  h.set(h.indexOf(w), w+2);
}
System.out.println(h);
```

9) Find the sum of all prices given in a String list
Example: List<String> myList = new List<String>{'$12.99', '$23.60', '$54.45'}; ==> 91.04

```
List<String> myList = new ArrayList<>();
myList.add("$12.99");
myList.add("$23.60");
myList.add("$54.45");
double sum = 0;

for(String w : myList){
  Double price = Double.valueOf(w.replace("$", ""));
  sum = sum + price;
}
System.out.println(sum);
```

10) Find the sum of the least and the greatest price given in a String list
Example: List<String> myList = new List<String>{'$12.99', '$8.25', '$23.60', '$54.45'}; ==> 62.70

```
List<String> myList = new ArrayList<>();
myList.add("$12.99");
myList.add("$23.60");
myList.add("$8.25");
myList.add("$54.45");
List<Double> decimalList = new ArrayList<>();

for(String w : myList){
  Double price = Double.valueOf(w.replace("$", ""));
  decimalList.add(price);
}
Collections.sort(decimalList);
Double sum = decimalList.get(0) + decimalList.get(decimalList.size()-1);
System.out.println(sum);
```

11) **Check if all elements are unique in an integer List by using loops.**

```
List<Integer> myList = new ArrayList<>();
myList.add(10);
myList.add(31);
myList.add(15);
myList.add(7);
myList.add(15);
myList.add(23);

Integer counter = 0;
for(Integer w : myList){
  for(Integer k : myList){
    if(w==k){
      counter++;
    }
  }
}
if(counter==myList.size()){
  System.out.println("There is no repeated element");
}else{
  System.out.println("At least one element was repeated");
}
```

12) Mountain Array ==> [0, 2, 5, 3, 1] It is constantly increasing to the maximum value, decreasing constantly after the maximum value

Not Mountain Array ==> [5, 2, 7, 1, 4] It is decreasing then increasing to the maximum value, decreasing and increasing after the maximum value

Type a code to check if any given array is Mountain Array or not.

```
int a[] = { 0, 2, 5, 3, 1 };
List<Integer> list = new ArrayList<>();
Arrays.stream(a).forEach(t -> list.add(t));

int max = list.stream().reduce(Integer.MIN_VALUE, Integer::max);
int idxOfMax = list.indexOf(max);

List<Integer> part1 = new ArrayList<>();
list.stream().filter(t -> list.indexOf(t) <= idxOfMax).forEach(t -> part1.add(t));

List<Integer> sortedPart1 = new ArrayList<>();
part1.stream().forEach(t -> sortedPart1.add(t));
Collections.sort(sortedPart1);

List<Integer> part2 = new ArrayList<>();
list.stream().filter(t -> list.indexOf(t) >= idxOfMax).forEach(t -> part2.add(t));

List<Integer> reverseSortedPart2 = new ArrayList<>();
part2.stream().forEach(t -> reverseSortedPart2.add(t));
Collections.sort(reverseSortedPart2);
Collections.reverse(reverseSortedPart2);

if (part1.equals(sortedPart1) && part2.equals(reverseSortedPart2)) {
        System.out.println(list + " is Mountain Array");
} else {
        System.out.println(list + " is not Mountain Array");
}
```

1) Type the code that shows how many different elements there are in a list.
Example: {10, 31, 15, 7, 15, 7, 7} ==> 4

```
List<Integer> myList = new ArrayList<>();
myList.add(10);
myList.add(31);
myList.add(15);
myList.add(7);
myList.add(15);
myList.add(7);
myList.add(7);

Set<Integer> mySet = new HashSet<>(myList);
System.out.println(mySet.size());
```

2) Type code to print different characters used in a String
Example: 'Mississippi' ⇒ Misp

```
String s = "Mississippi";
String[] arr = s.split("");
List<String> characters = Arrays.asList(arr);

Set<String> mySet = new HashSet<>(characters);
System.out.println(mySet);
```

3) Type code to print the common elements between a Set and a List

```
List<Integer> myList = new ArrayList<>();
myList.add(10);
myList.add(31);
myList.add(15);
myList.add(7);
myList.add(13);

Set<Integer> mySet = new HashSet<>();
mySet.add(10);
mySet.add(7);
mySet.add(35);
mySet.add(13);

mySet.retainAll(myList);
System.out.println(mySet);
```

4) Type code to print the different elements in a Set from a List

```
List<Integer> myList = new ArrayList<>();
myList.add(10);
myList.add(31);
myList.add(15);
myList.add(7);
myList.add(13);

Set<Integer> mySet = new HashSet<>();
mySet.add(10);
mySet.add(7);
mySet.add(35);
mySet.add(13);

mySet.removeAll(myList);
System.out.println(mySet);
```

5) Type code to check if a List has repeated elements or not

```
List<Integer> myList = new ArrayList<>();
myList.add(10);
myList.add(31);
myList.add(15);
myList.add(7);
myList.add(13);

Set<Integer> mySet = new HashSet<>(myList);

if(myList.size()>mySet.size()){
  System.out.println("The list has repeated elements");
}else{
  System.out.println("The list does not have repeated elements");
}
```

1) There is a Map which contains product names as key and number of the products as value. Type code to find the total number of products.

```
Map<String, Integer> product = new HashMap<>();
product.put("Laptop", 12);
product.put("TV", 53);
product.put("Refrigerator", 12);
product.put("Music System", 87);

Collection<Integer> numOfProducts = product.values();

Integer sum = 0;
for(Integer w : numOfProducts){
  sum = sum + w;
}
System.out.println(sum);
```

2) There is a Map which contains product names as key and number of the products as value. Type code to check if 'Laptop' is among the products.

```
Map<String, Integer> product = new HashMap<>();
product.put("Laptop", 12);
product.put("TV", 53);
product.put("Refrigerator", 12);
product.put("Music System", 87);

String expectedProduct = "Laptop";
if(product.containsKey(expectedProduct)){
  System.out.println(expectedProduct + " exists");
}else{
  System.out.println(expectedProduct + " does not exist");
}
```

3) **There is a Map which contains product names as key and number of the products as value. Print the prices in ascending order.**

```
Map<String, Integer> product = new HashMap<>();
product.put("Laptop", 82);
product.put("TV", 53);
product.put("Refrigerator", 12);
product.put("Music System", 87);
product.put("Mobile Phone", 53);

Object[] productPrices = product.values().toArray();
Arrays.sort(productPrices);

System.out.println(Arrays.toString(productPrices));
```

4) **There is a Map which contains product names as key and number of the products as value. Print the product names in alphabetical order.**

```
Map<String, Integer> product = new HashMap<>();
product.put("Laptop", 82);
product.put("TV", 53);
product.put("Refrigerator", 12);
product.put("Music System", 87);
product.put("Mobile Phone", 53);

Set<String> productNamesAsSet = product.keySet();

List<String> productNamesAsList = new ArrayList<>(productNamesAsSet);
Collections.sort(productNamesAsList);

System.out.println(productNamesAsList);
```

5) Type code to count the number of occurrences of the words in a String. (Case insensitive)

```
String s = "Apex is easy. Type codes to learn apex. To earn money learn apex.";
s = s.replaceAll("\\p{Punct}","").toLowerCase();
Map<String, Integer> wordOccurence = new HashMap<>();
String[] words = s.split(" ");

for(String w : words){
  Integer numOfOccurence = wordOccurence.get(w);
  if(numOfOccurence == null ){
     wordOccurence.put(w, 1);
  }else{
     wordOccurence.put(w, numOfOccurence+1);
  }
}
System.out.println(wordOccurence);
```

6) How to check the number of repeated elements in a List

```
List<Integer> myList = new ArrayList<>();
myList.add(12);
myList.add(21);
myList.add(12);
myList.add(13);
myList.add(12);
myList.add(21);
myList.add(35);
Map<Integer, Integer> myMap = new HashMap<>();

for(Integer w : myList){
  Integer numOfOccurence = myMap.get(w);
  if(numOfOccurence==null){
     myMap.put(w, 1);
  }else{
     myMap.put(w, numOfOccurence+1);
  }
}
Collection<Integer> myValues = myMap.values();
Integer counter = 0;
for(Integer w : myValues){
  if(w>1){
     counter++;
  }
}
System.out.println("There are " + counter + " repeated elements");
```