[2CEIT5PE5: MOBILE APPLICATION DEVELOPMENT]

Practical: 1

AIM - Kotlin Programs

Submitted By: Chauhan Parth Dineshkumar
Enrollment number: 22012012041

Ganpat University
॥ विद्यया समाजोत्कर्षः ॥
U.V. Patel College of Engineering

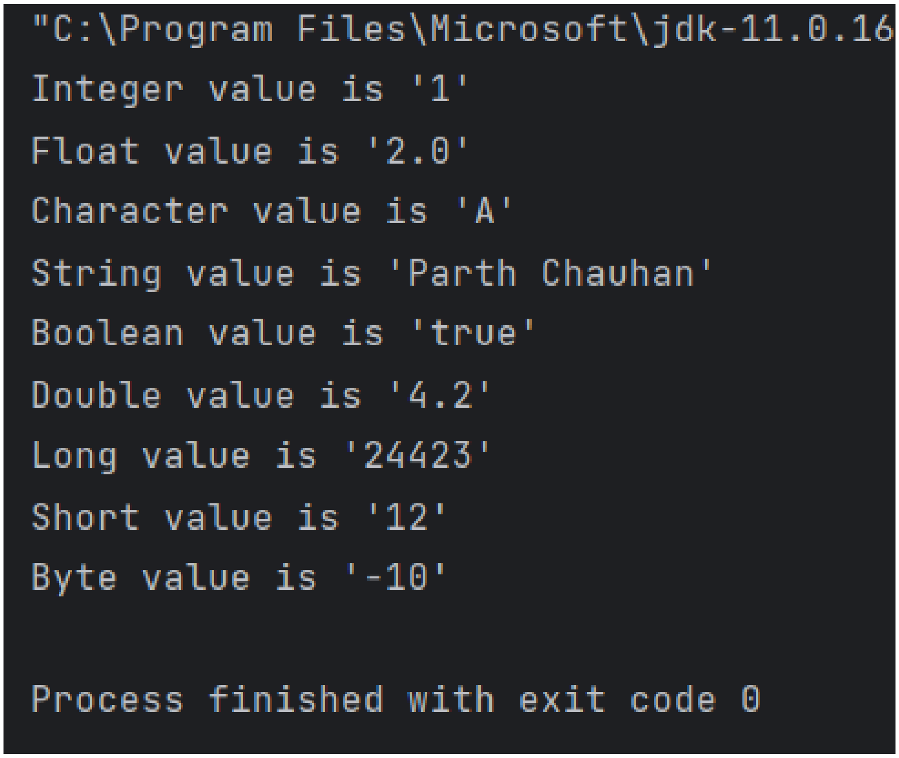
Department of Computer Engineering/Information Technology

1.1 Store & Display values in different variable of different type (Integer, Double, Float, Long, Short, Byte, Char, Boolean, String)

Answer:

```
fun main()
{
   val a:Int=1
   val b:Char='A'
   val c:Float=2.0f
   val d:String="Parth Chauhan"
   val e:Boolean=true
   val f:Double=4.2
   val g:Long=24423
   val h:Short=12
   val i:Byte=-10
   println("Integer value is '$a'")
   println("Float value is '$c'")
   println("Character value is '$b'")
   println("String value is '$d'")
   println("Boolean value is '$e'")
   println("Double value is '$f'")
   println("Long value is '$g'")
   println("Short value is '$h'")
   println("Byte value is '$i'")
}
```

Output:



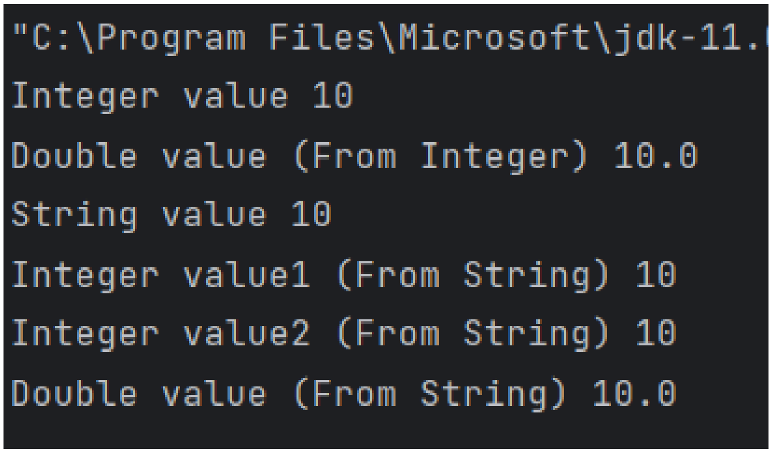
```
"C:\Program Files\Microsoft\jdk-11.0.16
Integer value is '1'
Float value is '2.0'
Character value is 'A'
String value is 'Parth Chauhan'
Boolean value is 'true'
Double value is '4.2'
Long value is '24423'
Short value is '12'
Byte value is '-10'

Process finished with exit code 0
```

1.2 Type conversion: Integer to Double, String to Integer, String to Double.

Answer:

```
fun main()
{
   val num:Int=10
   val num1:Double=num.toDouble()
   val num2:String="10"
   val num3:Int=num2.toInt()
   val num4:Int=num2.toInt()
   val num5:Double=num2.toDouble()
   println("Integer value $num")
   println("Double value (From Integer) $num1")
   println("String value $num2")
   println("Integer value1 (From String) $num3")
   println("Integer value2 (From String) $num4")
   println("Double value (From String) $num5")
}
```

Output:



```
"C:\Program Files\Microsoft\jdk-11.
Integer value 10
Double value (From Integer) 10.0
String value 10
Integer value1 (From String) 10
Integer value2 (From String) 10
Double value (From String) 10.0
```

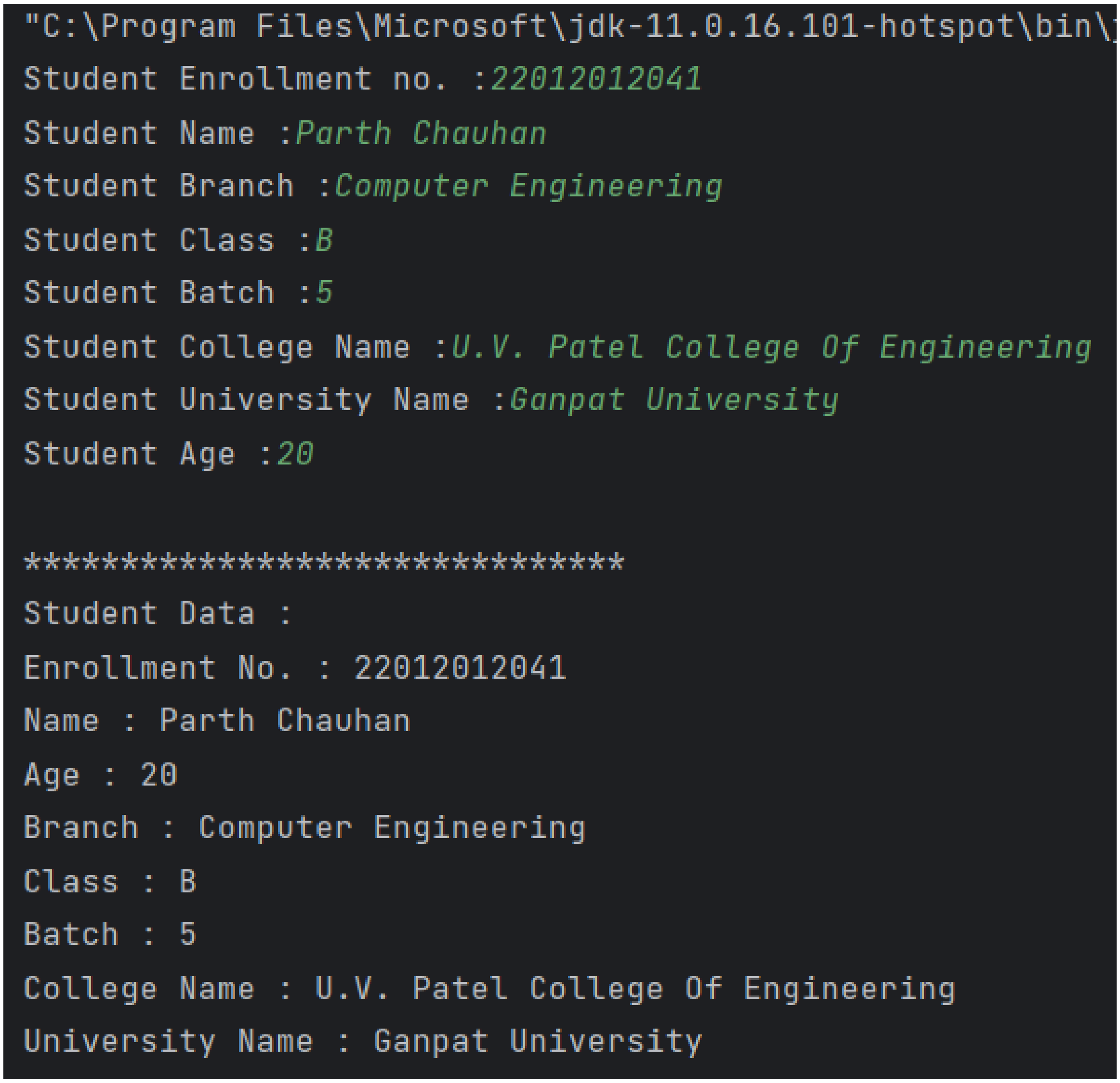
1.3 Scan student's information and display all the data

Answer:

```
fun main()
{
   print("Student Enrollment no. :")
   val erno= readln()
   print("Student Name :")
   val name= readln()
   print("Student Branch :")
   val branch= readln()
   print("Student Class :")
   val cls= readln()
   print("Student Batch :")
   val batch= readln()
   print("Student College Name :")
   val clg= readln()
   print("Student University Name :")
   val uni= readln()
   print("Student Age :")
   val age= readln()
   println()
   println("*******************************")
   println("Student Data : ")
   println("Enrollment No. : $erno")
```

```
        println("Name : $name")
        println("Age : $age")
        println("Branch : $branch")
        println("Class : $cls")
        println("Batch : $batch")
        println("College Name : $clg")
        println("University Name : $uni")
    }
```

Output:



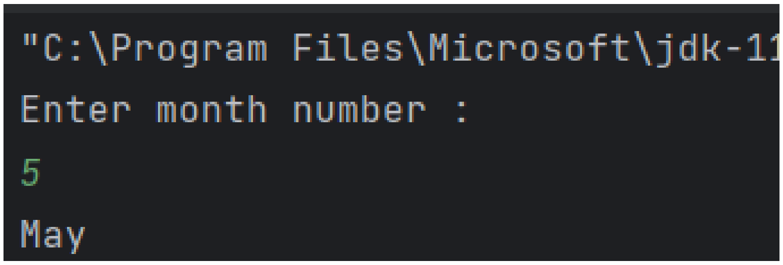
```
"C:\Program Files\Microsoft\jdk-11.0.16.101-hotspot\bin\
Student Enrollment no. :22012012041
Student Name :Parth Chauhan
Student Branch :Computer Engineering
Student Class :B
Student Batch :5
Student College Name :U.V. Patel College Of Engineering
Student University Name :Ganpat University
Student Age :20

*******************************
Student Data :
Enrollment No. : 22012012041
Name : Parth Chauhan
Age : 20
Branch : Computer Engineering
Class : B
Batch : 5
College Name : U.V. Patel College Of Engineering
University Name : Ganpat University
```

1.4 Find the number is odd or even by using Control Flow inside println() method.

Answer:

```
fun main()
{
   println("Enter number : ")
   var a= readLine()!!.toInt()
   if (a%2==0){
      print("${a} is Even")
   }
   else
   {
      print("${a} is Odd")
   }
}
```

Output:



```
"C:\Program Files\Microsoft\jdk-11
Enter month number :
5
May
```

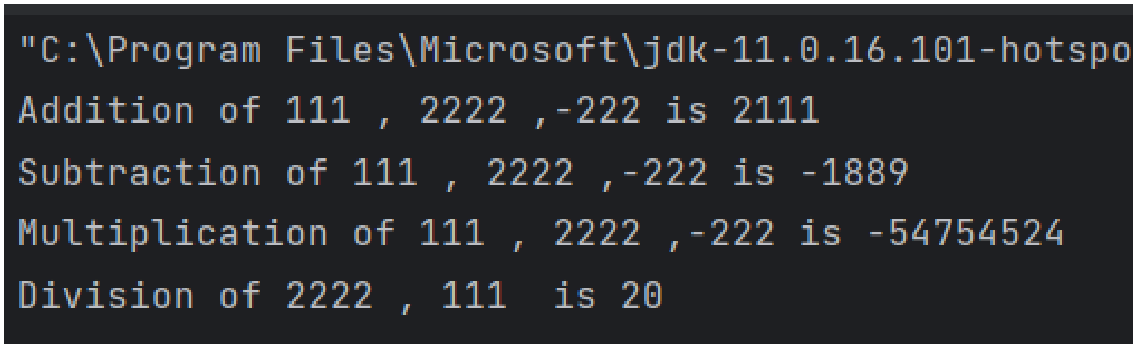
1.5 Display month name using When

Answer:

```
fun main()
{
   println("Enter month number : ")
   var month= readLine()!!.toInt()
   when(month){
      1-> print("January")
      2-> print("February")
      3-> print("March")
      4-> print("April")
      5-> print("May")
```

```
        6-> print("June")
        7-> print("July")
        8-> print("August")
        9-> print("September")
        10-> print("October")
        11-> print("November")
        12-> print("December")
    }
}
```

Output:



```
"C:\Program Files\Microsoft\jdk-11.0.16.101-hotspo
Addition of 111 , 2222 ,-222 is 2111
Subtraction of 111 , 2222 ,-222 is -1889
Multiplication of 111 , 2222 ,-222 is -54754524
Division of 2222 , 111  is 20
```

1.6 By using a user defined function perform all arithmetic operations.

Answer:

```
fun add(n1:Int,n2:Int,n3:Int){
    println("Addition of ${n1} , ${n2} ,${n3} is ${n1+n2+n3}");
}
fun div(n1:Int,n2:Int){
    println("Division of ${n1} , ${n2}  is ${n1/n2}");
}
fun sub(n1:Int,n2:Int,n3:Int){
    println("Subtraction of ${n1} , ${n2} ,${n3} is ${n1-n2-n3}");
}
fun mul(n1:Int,n2:Int,n3:Int){
    println("Multiplication of ${n1} , ${n2} ,${n3} is ${n1*n2*n3}");
}
fun main()
{
    add(111,2222,-222);
    sub(111,2222,-222);
    mul(111,2222,-222);
    div(2222,111);

}
```

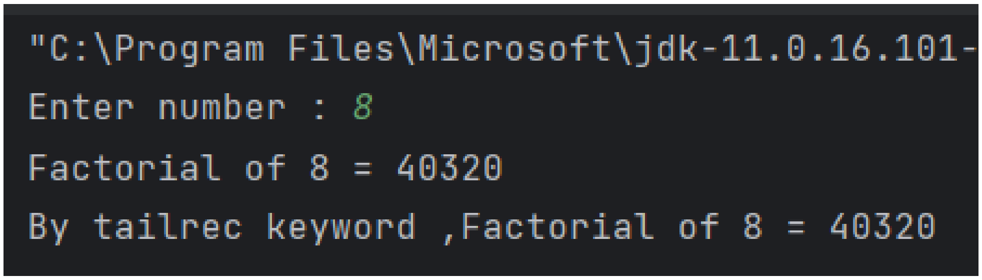
Output:

1.7 Find the factorial of number by recursion. Explain "tailrec" keyword.

Answer:

```
fun fact(num:Int):Int{
    if(num==1){
        return num;
    }
    else return num*fact(num-1);
}

tailrec fun factorial(n: Int, run: Int = 1): Long {
    return if (n == 1){
        run.toLong()
    } else {
        factorial(n-1, run*n)
    }
}
fun main()
{
    print("Enter number : ")
    var num= readln()!!.toInt();
    var ans=fact(num);
    var tail=factorial(5);
    println("Factorial of ${num} = ${ans}")
    println("By tailrec keyword ,Factorial of ${num} = ${ans}")
}
```

Output:



```
"C:\Program Files\Microsoft\jdk-11.0.16.101-
Enter number : 8
Factorial of 8 = 40320
By tailrec keyword ,Factorial of 8 = 40320
```

1.8 Create different types of Array as shown in image. Explore Arrays.deepToString(), contentDeepToString() methods, IntArray variable .joinToString() and use in program to print Array. Explore range, downTo, until etc. for loop and use in this program. Sort Array of Integer data type without using inbuilt function & with using inbuilt function

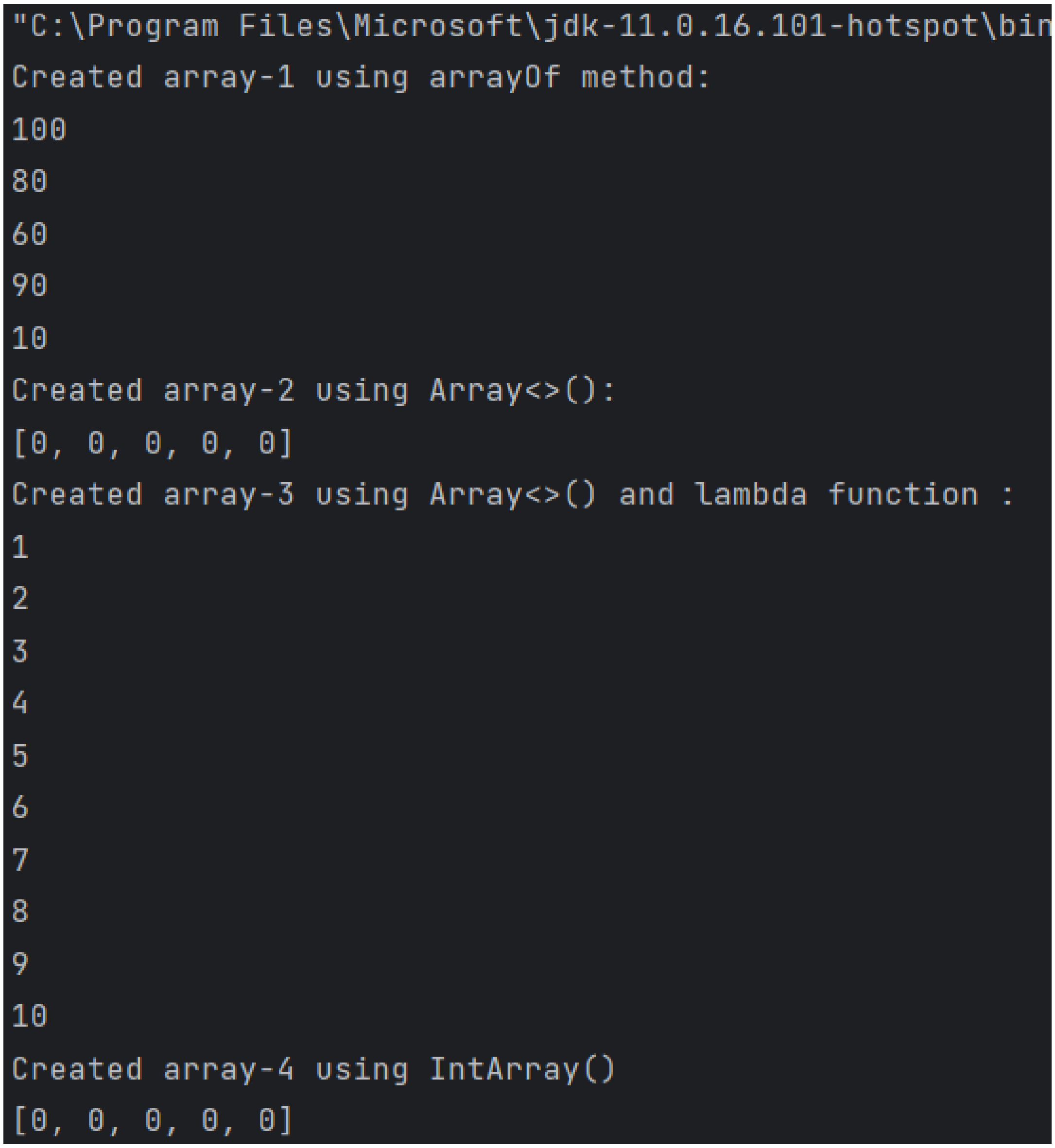
Answer:

```
import java.util.*
import kotlin.collections.ArrayList

fun printArray(a:Array<Int>){
    for (item in a.size-1 downTo 0) {
        println(a[item])
    }
}
fun selectionSort(arr: IntArray) {
    for (i in 0 until arr.size - 1) {
        var minIndex = i
        for (j in i + 1 until arr.size) {
            if (arr[j] < arr[minIndex]) {
                minIndex = j
            }
        }
        if (minIndex != i) {
            val temp = arr[i]
            arr[i] = arr[minIndex]
            arr[minIndex] = temp
        }
    }
}
```

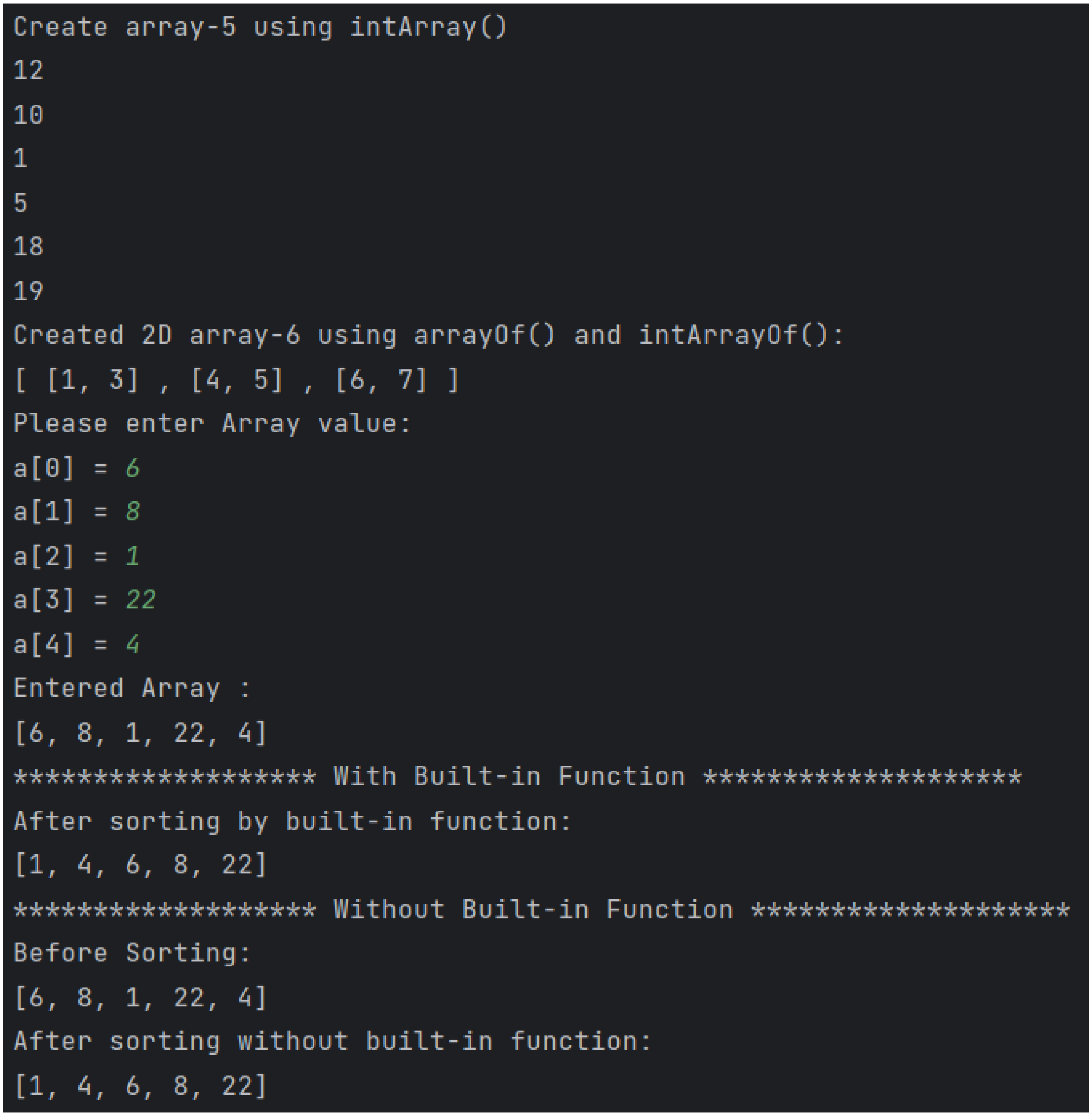
```
fun main() {
  var a1 = arrayOf(10, 90, 60, 80, 100);
  println("Created array-1 using arrayOf method: ")
  printArray(a1);

  var a2=Array<Int>(5){0};
  println("Created array-2 using Array<>(): ");
  println(Arrays.deepToString(a2));

  var a3=Array<Int>(10){i:Int->i+1};
  println("Created array-3 using Array<>() and lambda function :");
  for (num in 1.rangeTo(10)){
    println(num);
  }

  var a4=IntArray(5);
  println("Created array-4 using IntArray()");

  println(a4.contentToString());

  var a5= intArrayOf(12,10,1,5,18,19);
  println("Create array-5 using intArray()");
  for (i in 0 until a5.size){
    println(a5[i]);
  }

  val a6= arrayOf(intArrayOf(1,3), intArrayOf(4,5), intArrayOf(6,7));
  println("Created 2D array-6 using arrayOf() and intArrayOf():");
  print("[ ")
  for (i in a6.indices) {
    print("[")
    for (j in a6[i].indices) {
      print(a6[i][j])
      if (j < a6[i].size - 1) {
        print(", ")
      }
    }

    print("]")
```

```
          if (i < a6.size - 1) {
             print(" , ")
          }
       }
       println(" ]")

       println("Please enter Array value: ");
       var a7=IntArray(5);
          for (i in a7.indices ){
          print("a[${i}] = ");
          a7[i]= readLine()?.toIntOrNull()!!;
       }
       println("Entered Array :")
       println(a7.contentToString());

       println("****************** With Built-in Function
********************");
       println("After sorting by built-in function:");
       var a8=a7.sorted();

       println(a8)

       println("****************** Without Built-in Function
********************");
       println("Before Sorting:");
       println(a7.contentToString());

       println("After sorting without built-in function:");
       selectionSort(a7);
       println(a7.contentToString());
    }
```

Output:



```
"C:\Program Files\Microsoft\jdk-11.0.16.101-hotspot\bin
Created array-1 using arrayOf method:
100
80
60
90
10
Created array-2 using Array<>():
[0, 0, 0, 0, 0]
Created array-3 using Array<>() and lambda function :
1
2
3
4
5
6
7
8
9
10
Created array-4 using IntArray()
[0, 0, 0, 0, 0]
```



```
Create array-5 using intArray()
12
10
1
5
18
19
Created 2D array-6 using arrayOf() and intArrayOf():
[ [1, 3] , [4, 5] , [6, 7] ]
Please enter Array value:
a[0] = 6
a[1] = 8
a[2] = 1
a[3] = 22
a[4] = 4
Entered Array :
[6, 8, 1, 22, 4]
******************* With Built-in Function ********************
After sorting by built-in function:
[1, 4, 6, 8, 22]
******************* Without Built-in Function ********************
Before Sorting:
[6, 8, 1, 22, 4]
After sorting without built-in function:
[1, 4, 6, 8, 22]
```

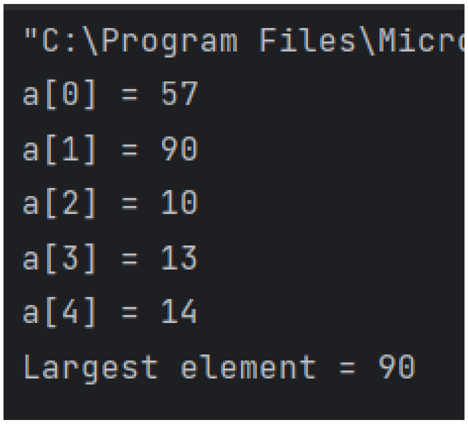
1.9 Find the max number from ArrayList

Answer:

```
fun main()
{
   val num=ArrayList<Int>();
   num.add(57)
```

```
    num.add(90)
    num.add(10)
    num.add(13)
    num.add(14)
    for (i in num.indices){
        println("a[${i}] = ${num[i]}")
    }
    var max=num[0];
    for (i in 1..num.size-1){
        if (max<num[i]) max=num[i]
    }
    println("Largest element = ${max}")
}
```

Output:



```
"C:\Program Files\Micro
a[0] = 57
a[1] = 90
a[2] = 10
a[3] = 13
a[4] = 14
Largest element = 90
```

1.10 Write Different types of Class & Constructor. Create a class Car and set various members like type, model, price, owner, milesDrive. add the function getCarPrice in it. Create an object of Car class and access property of it. (getCarInformation(), getOriginalCarPrice(), getCurrentCarPrice(), displayCarInfo() etc.)

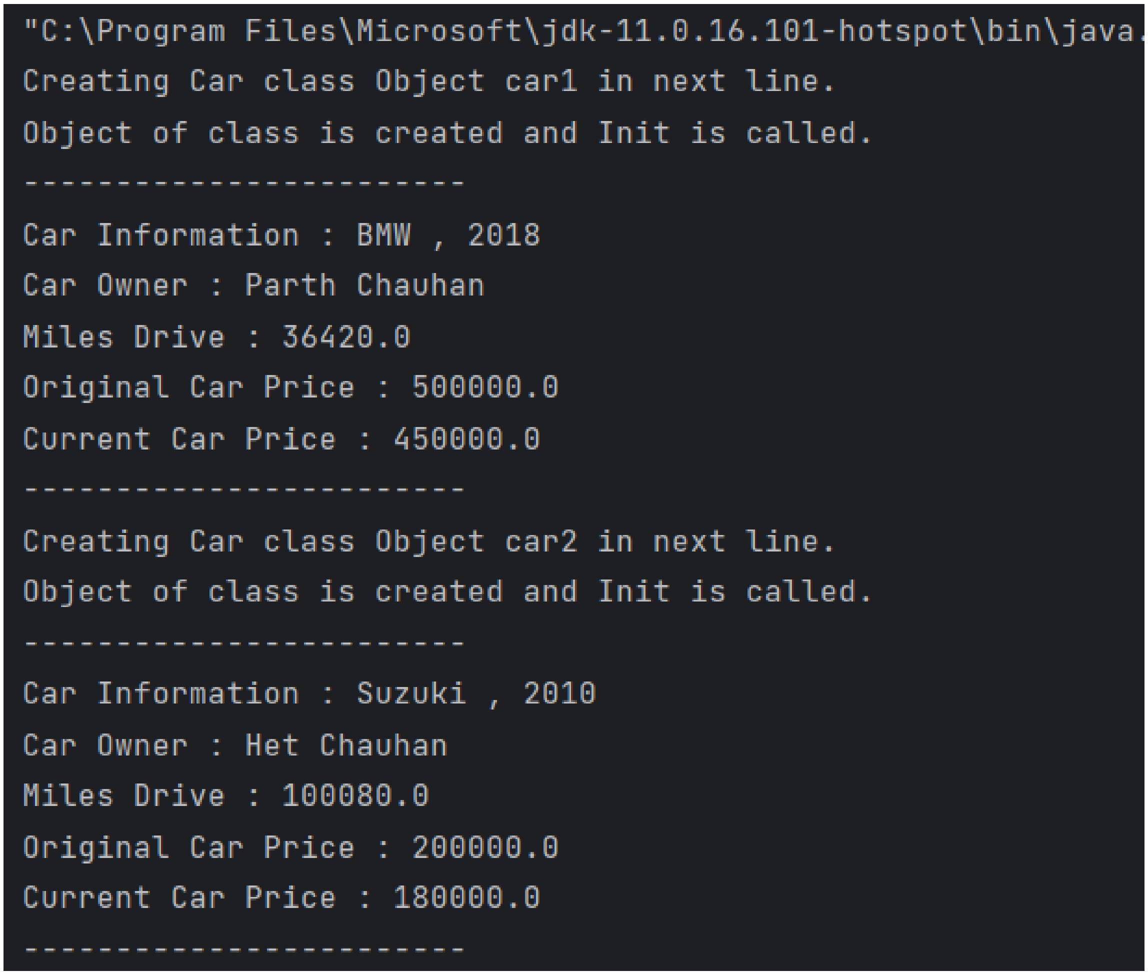
Answer:

```
class Car(type:String, model:Int, price: Double, owner:String, milesDrive:
Double){
    var carType=type;
    val carModel=model;
    val carPrice=price;
```

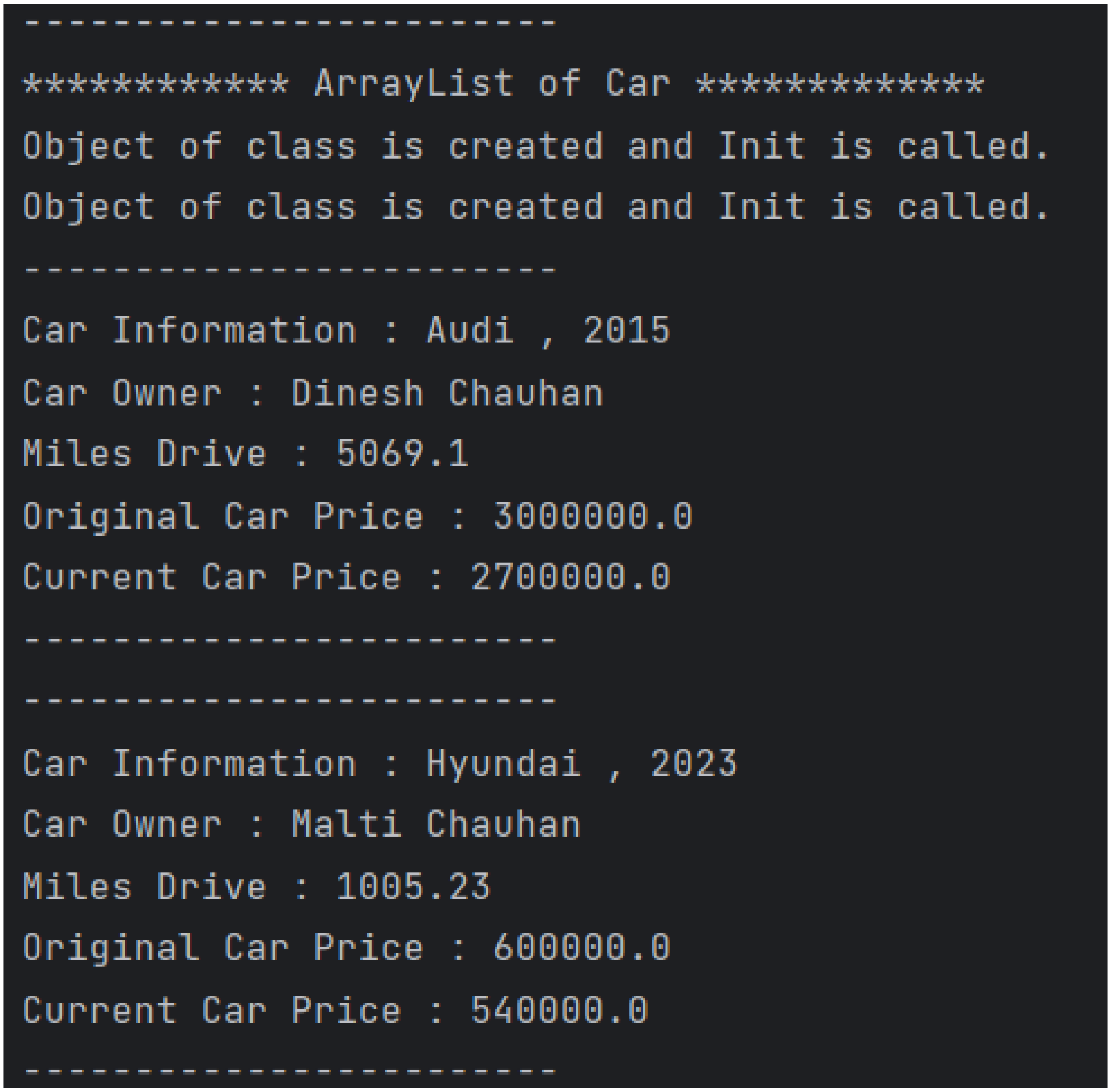
```
  var carOwner=owner;
  val carMilesDrive=milesDrive;
  init{
    println("Object of class is created and Init is called.");
  }
  fun getCarInformation(){
    println("Car Information : ${carType} , ${carModel}");
    println("Car Owner : ${carOwner}");
    println("Miles Drive : ${carMilesDrive}");
  }

  fun getOriginalCarPrice():Double{
    return carPrice;
  }

  fun getCurrentCarPrice():Double{
    return carPrice-(carPrice*0.1);
  }
  fun displayCarInfo(){
    println("------------------------");
    getCarInformation();
    println("Original Car Price : ${getOriginalCarPrice()}");
    println("Current Car Price : ${getCurrentCarPrice()}");
    println("------------------------");
  }
}

fun main()
{
  println("Creating Car class Object car1 in next line.");
  var c1=Car("BMW",2018,500000.0,"Parth Chauhan",36420.0);
  c1.displayCarInfo();
  println("Creating Car class Object car2 in next line.");
  var c2=Car("Suzuki",2010,200000.0,"Het Chauhan",100080.0);
  c2.displayCarInfo();

  println("************ ArrayList of Car *************")
  var l= ArrayList<Car>();
  l.add(Car("Audi",2015,3000000.0,"Dinesh Chauhan",5069.10))
```

```
        l.add(Car("Hyundai",2023,600000.0,"Malti Chauhan",1005.23))
        for(i in l){
           i.displayCarInfo();
        }
    }
```

Output:



```
"C:\Program Files\Microsoft\jdk-11.0.16.101-hotspot\bin\java.
Creating Car class Object car1 in next line.
Object of class is created and Init is called.
------------------------
Car Information : BMW , 2018
Car Owner : Parth Chauhan
Miles Drive : 36420.0
Original Car Price : 500000.0
Current Car Price : 450000.0
------------------------
Creating Car class Object car2 in next line.
Object of class is created and Init is called.
------------------------
Car Information : Suzuki , 2010
Car Owner : Het Chauhan
Miles Drive : 100080.0
Original Car Price : 200000.0
Current Car Price : 180000.0
------------------------
```



```
------------------------
************ ArrayList of Car *************
Object of class is created and Init is called.
Object of class is created and Init is called.
------------------------
Car Information : Audi , 2015
Car Owner : Dinesh Chauhan
Miles Drive : 5069.1
Original Car Price : 3000000.0
Current Car Price : 2700000.0
------------------------
------------------------
Car Information : Hyundai , 2023
Car Owner : Malti Chauhan
Miles Drive : 1005.23
Original Car Price : 600000.0
Current Car Price : 540000.0
------------------------
```

1.11 Write about Operator Overloading. Perform Matrix Addition, Subtraction & Multiplication using Class Matrix & operator overloading. Overload toString() function in Matrix class.

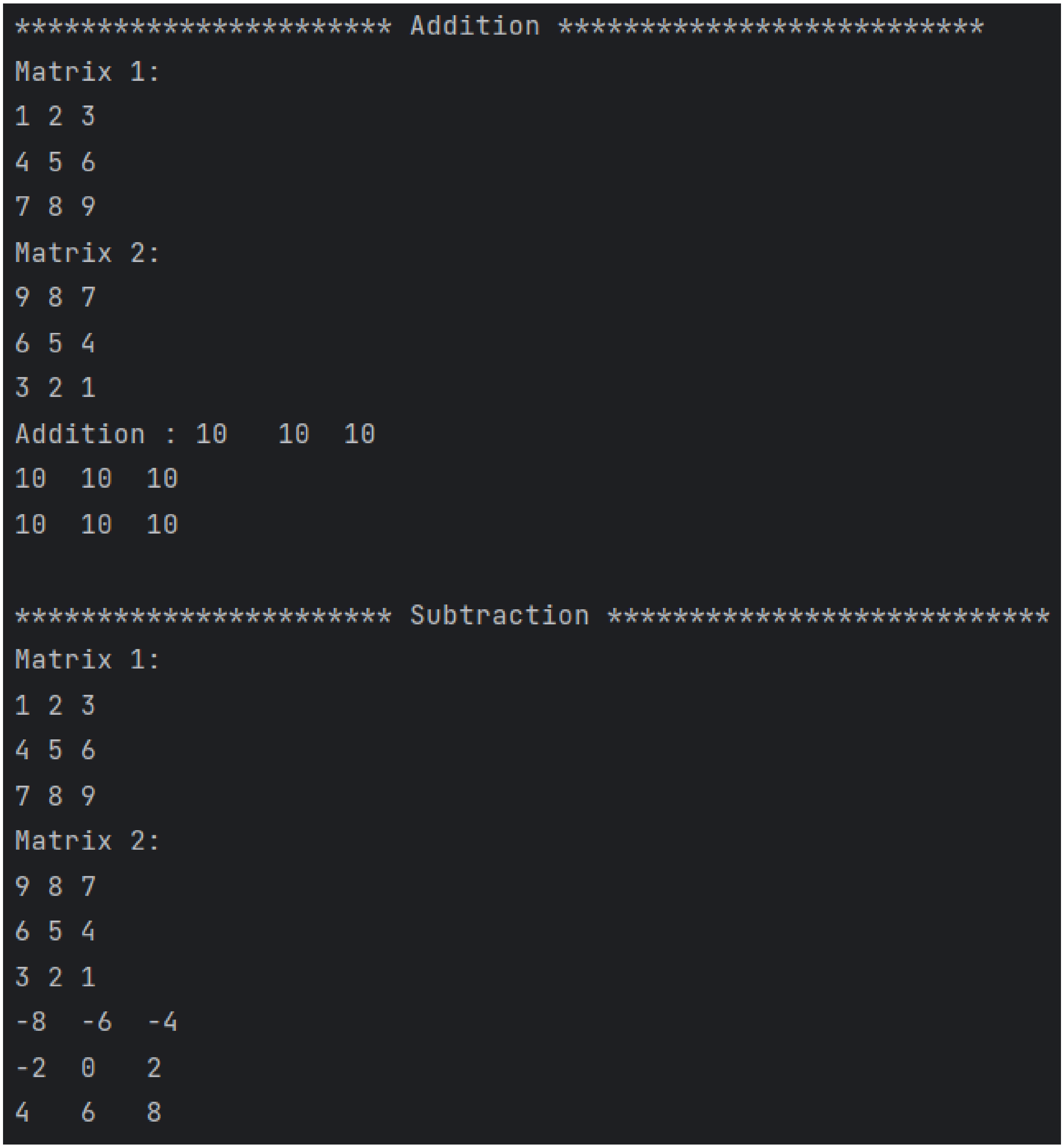
Answer:

```
class Matrix(private val rows: Int, private val cols: Int) {
    val data: Array<Array<Int>> = Array(rows) { Array(cols) { 0 } }

    fun showMatrix(a:Matrix){
        for(i in 0 until rows){
            for (j in 0 until cols){
```

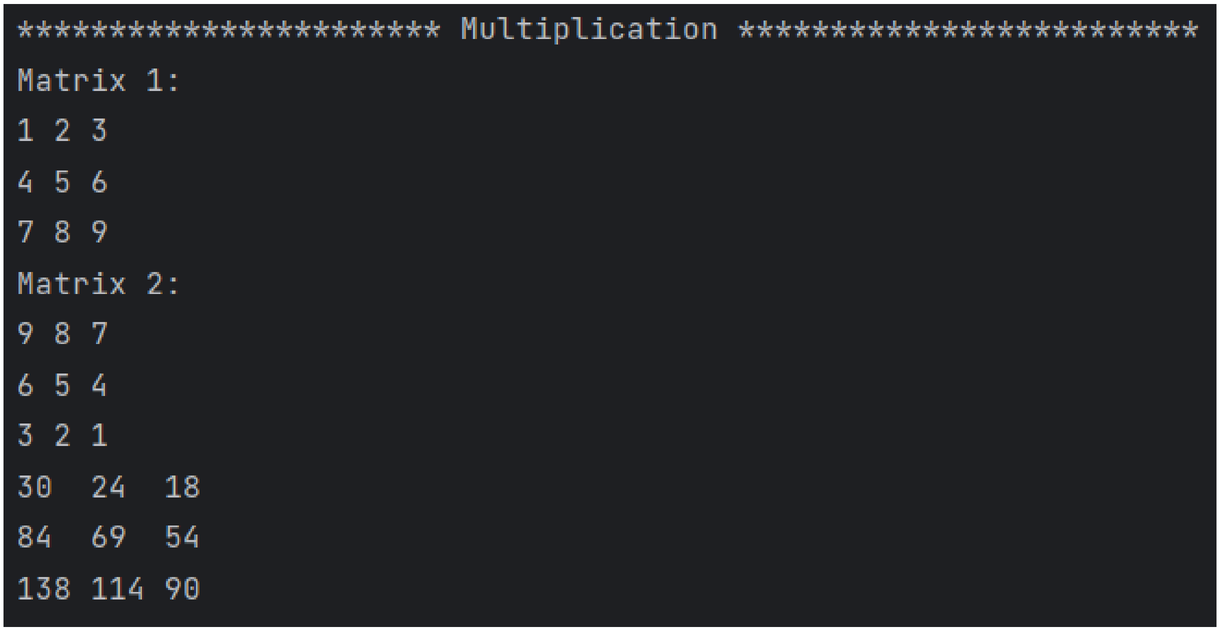
```
        print("${a.data[i][j]} ");
      }
      println();
    }
  }

  operator fun plus(other: Matrix): Matrix {
    require(rows == other.rows && cols == other.cols) { "Matrix
dimensions must be the same for addition." }

    val result = Matrix(rows, cols)
    for (i in 0 until rows) {
      for (j in 0 until cols) {
        result.data[i][j] = this.data[i][j] + other.data[i][j]
      }
    }
    return result
  }

  operator fun minus(other: Matrix): Matrix {
    require(rows == other.rows && cols == other.cols) { "Matrix
dimensions must be the same for subtraction." }

    val result = Matrix(rows, cols)
    for (i in 0 until rows) {
      for (j in 0 until cols) {
        result.data[i][j] = this.data[i][j] - other.data[i][j]
      }
    }
    return result
  }

  operator fun times(other: Matrix): Matrix {
    require(cols == other.rows) { "Number of columns in the first
matrix must be equal to the number of rows in the second matrix for
multiplication." }

    val result = Matrix(rows, other.cols)
    for (i in 0 until rows) {
        for (j in 0 until other.cols) {
          for (k in 0 until cols) {
            result.data[i][j] += this.data[i][k] * other.data[k][j]
          }
        }
      }
      return result
    }

    override fun toString(): String {
      val sb = StringBuilder()
      for (i in 0 until rows) {
        for (j in 0 until cols) {
          sb.append("${data[i][j]}\t")
        }
        sb.append("\n")
      }
      return sb.toString()
    }
  }
  fun main() {
    val matrix1 = Matrix(3, 3)
    matrix1.apply {
      data[0][0] = 1
      data[0][1] = 2
      data[0][2] = 3
      data[1][0] = 4
      data[1][1] = 5
      data[1][2] = 6
      data[2][0] = 7
      data[2][1] = 8
      data[2][2] = 9
    }

    val matrix2 = Matrix(3, 3)
    matrix2.apply {
      data[0][0] = 9
      data[0][1] = 8
      data[0][2] = 7
```

```
            data[1][0] = 6
            data[1][1] = 5
            data[1][2] = 4

            data[2][0] = 3
            data[2][1] = 2
            data[2][2] = 1
        }

        println("*********************** Addition **************************");
        val sumMatrix = matrix1 + matrix2
        println("Matrix 1:")
        matrix1.showMatrix(matrix1);
        println("Matrix 2:")
        matrix2.showMatrix(matrix2);
        println("Addition : ${sumMatrix}");


        println("********************** Subtraction *************************");
        val diffMatrix = matrix1 - matrix2
        println("Matrix 1:")
        matrix1.showMatrix(matrix1);
        println("Matrix 2:")
        matrix2.showMatrix(matrix2);
        println(diffMatrix)


        val productMatrix = matrix1 * matrix2
        println("********************* Multiplication ***********************");
        println("Matrix 1:")
        matrix1.showMatrix(matrix1);
        println("Matrix 2:")
        matrix2.showMatrix(matrix2);
        println(productMatrix)
    }
```

Output:



```
*********************** Addition **************************
Matrix 1:
1 2 3
4 5 6
7 8 9
Matrix 2:
9 8 7
6 5 4
3 2 1
Addition : 10   10  10
10  10  10
10  10  10

*********************** Subtraction ***************************
Matrix 1:
1 2 3
4 5 6
7 8 9
Matrix 2:
9 8 7
6 5 4
3 2 1
-8  -6  -4
-2  0   2
4   6   8
```



```
*********************** Multiplication *************************
Matrix 1:
1 2 3
4 5 6
7 8 9
Matrix 2:
9 8 7
6 5 4
3 2 1
30  24  18
84  69  54
138 114 90
```