Assignment(1~4)

Factorial(#1)

Problem

- Write a program that prints factorials from 1 to the given number.
- The program should run like this!

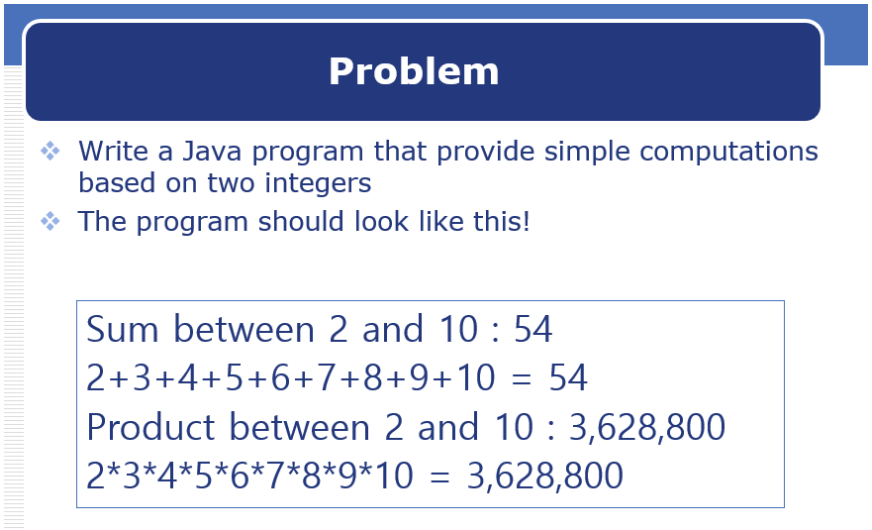
```
Factorial of 1: 1=1
Factorial of 2: 1*2=2
Factorial of 3: 1*2*3=6
Factorial of 4: 1*2*3*4=24
Factorial of 5: 1*2*3*4*5=120
Factorial of 6: 1*2*3*4*5*6=720
Factorial of 7: 1*2*3*4*5*6*7=5040
Factorial of 8: 1*2*3*4*5*6*7*8=40320
Factorial of 9: 1*2*3*4*5*6*7*8*9=362880
Factorial of 10: 1*2*3*4*5*6*7*8*9*10=3628800
```

```
public class FactorialMain {
  public static void main(String[] args) {
    for (int i = 1; i <= 10; i ++) {
      System.out.print("\nFactorial of " + i + ": ");
      for (int j = 1; j <= i; j ++) {
        if (j == 1) System.out.print(1);
        else System.out.print("*" + j);
      }
      System.out.print("=");
      System.out.print(factorial(i));
    }

  }

  public static long factorial(final int n) {
    if (n <= 1) {
      return n;
    }
    else {
      return factorial(n - 1)*n;
    }
  }
}
```

Math(#2)




```
public class MathMain {
  public static void main(String[] args) {
    int begin = 2;
    int end = 10;

    long sum = getSumBetween(begin, end);
    System.out.printf("Sum between %d and %d: %d\n", begin, end, sum);
    printSumBetween(begin, end, sum);

    long product = getProductBetween(begin, end);
    System.out.printf("Product between %d and %d: %d\n", begin, end, product);
    printProductBetween(begin, end, product);

  }

  public static long getSumBetween(int begin, int end) {
    long sum = 0;
    for (int i = begin; i < end + 1; i++) {
      sum += i;
    }

    return sum;
  }

  public static void printSumBetween(int begin, int end, long sum) {
    for (int i = begin; i < end + 1; i ++) {
      if (i == begin) System.out.print(i);
      else System.out.print("+" + i);
```

```
    }
    System.out.printf(" = %d\n", sum);
  }

  public static long getProductBetween(int begin, int end) {
    long product = 1;
    for (int i = begin; i < end + 1; i++) {
      product *= i;
    }

    return product;
  }

  public static void printProductBetween(int begin, int end, long product) {
    for (int i = begin; i < end + 1; i ++) {
      if (i == begin) System.out.print(i);
      else System.out.print("*" + i);
    }
    System.out.printf(" = %d\n", product);
  }
}
```

ArrayEnum(#3)

Problem

- Write a Java program based on the given numbers
- The program should run like this!
- Commands are not case sensitive.
- The list contains no more than 100 values.

ADD 10
add 20
LIST
10 20
Sum
30
Add 30
suM
60
Sum
60
Summ
Invalid Command
list
10 20 30
Quit
Bye!

```
import java.util.Scanner;

enum Command {
  ADD,
  LIST,
  SUM,
  INVALID,
  QUIT
};
public class ArrayEnum {
  public static void main(String[] args) {
    int index = 0;
    int[] values = new int[100];


    final Scanner scanner = new Scanner(System.in);

    while ( true ) {
      final Command command = getCommand(scanner);
      if ( command == Command.QUIT ) {
        System.out.println("Bye!");
        break;
      }
      switch ( command ) {
      case ADD:
        final int newValue = getValue(scanner);
        values[index] = newValue;
        index++;
        break;
      case LIST:
        printList(values, index);
        break;
      case SUM:
        System.out.println(getSum(values, index));
        break;
      case INVALID:
        System.out.println("Ivalid Command");
        default: break;
      }
    }
    scanner.close();
  }

  // Input
  public static Command getCommand(Scanner scan) {
    String cmd = scan.next();
    cmd = cmd.toUpperCase();

    Command command;
    switch (cmd) {
    case "ADD":
      command = Command.ADD;
```

```
      break;
    case "LIST":
      command = Command.LIST;
      break;
    case "SUM":
      command = Command.SUM;
      break;
    case "QUIT":
      command = Command.QUIT;
      break;
    default:
      command = Command.INVALID;
      break;
    }
    return command;
  }

  // LIST
  public static void printList(int[] pList, int pIndex) {
    for(int i = 0; i < pIndex; i++) {
      if (i == (pIndex - 1)) {
        System.out.printf("%d\n", pList[i]);
      }
      else {
        System.out.printf("%d ", pList[i]);
      }
    }
  }

  // ADD
  public static int getValue(Scanner scan) {
    int val = scan.nextInt();

    return val;
  }

  // SUM
  public static int getSum(int[] pList, int pIndex) {
    int sum = 0;
    for(int i = 0; i < pIndex; i++) {
      sum += pList[i];
    }

    return sum;
  }

  // Your code here
}
```

Control Structure(#4)

Problem

- Commands are not case sensitive.
- The set contains no more than 100 values.

```
add Hello
Element Size: 1, Values = Hello
Add Java
Element Size: 2, Values = Hello, Java
add hello
Element Size: 3, Values = Hello, Java, hello
aDD Java
Element Size: 3, Values = Hello, Java, hello
remove Hello
Element Size: 2, Values = Java, hello
remove java
Element Size: 2, Values = Java, hello
add Good
Element Size: 3, Values = Good, Java, hello
remove Java
Element Size: 2, Values = Good, hello
clean
Element Size: 0, Values =
add PNU
Element Size: 1, Values = PNU
add is
Element Size: 2, Values = PNU, is
add Wonderful
Element Size: 3, Values = PNU, is, Wonderful
quit
BYE!
```

```
import java.util.Scanner;

enum StringCommand {
  ADD,
  REMOVE,
  CLEAN,
  QUIT,
  INVALID
};

public class StringSetManager {
  static int index = 0;
  public static void main(String[] args) {
    final Scanner scanner = new Scanner(System.in);
    String[] stringSet = new String[100];
    while (true) {
      final StringCommand command = getCommand(scanner);
      if (command == StringCommand.QUIT) {
        System.out.println("BYE!"); break;
      }
      switch (command) {
      case ADD: {
        final String str = getString(scanner);
        executeAdd(stringSet, str);
        break;
      }
      case REMOVE: {
```

```
        final String str = getString(scanner);
        executeRemove(stringSet, str);
        break;
      }
      case CLEAN: {
        executeClear(stringSet);
        break;
      }
      default:
        System.out.println("Unknown Command!");
        break;
      }
      executePrint(stringSet);
    }
  }

  // getCommand
  public static StringCommand getCommand(Scanner scan) {
    String cmd = scan.next();
    cmd = cmd.toUpperCase();

    StringCommand command;
    switch (cmd) {
    case "ADD":
      command = StringCommand.ADD;
      break;
    case "REMOVE":
      command = StringCommand.REMOVE;
      break;
    case "CLEAN":
      command = StringCommand.CLEAN;
      break;
    case "QUIT":
      command = StringCommand.QUIT;
      break;
    default:
      command = StringCommand.INVALID;
      break;
    }
    return command;
  }

  // getString
  public static String getString(Scanner scan) {
    String str = scan.next();
    return str;
  }

  // execute add
  public static void executeAdd(String[] stringSet, String str) {
    stringSet[index] = str;
    index++;
  }
```

```
  // execute remove
  public static void executeRemove(String[] stringSet, String str) {
    for (int i = 0; i < index; i++) {
      if (stringSet[i].equals(str)) {
        index--;
        for (int j = i; j < index; j++) {
          stringSet[j] = stringSet[j+1];
        }
        break;
      }
    }
  }

  // execute clear
  public static void executeClear(String[] stringSet) {
    for (int i = 0; i < index; i++) {
      stringSet[i] = null;
    }
    index = 0;
  }

  // execute print
  public static void executePrint(String[] stringSet) {
    System.out.printf("Element Size: %d, Values = ", index);
    if (index != 0) {
      for (int i = 0; i < index; i++) {
        if (i == index - 1) {
          System.out.printf("%s\n", stringSet[i]);
        }
        else {
          System.out.printf("%s, ", stringSet[i]);
        }
      }
    }
    else {
      System.out.printf("\n");
    }
  }
}
```