NAME	:	FARICHA AULIA
NIM	:	2141720155
ABSENT	:	08
CLASS	:	1I - IT
TOPIC	:	JOBSHEET 4 BF AND DC

4.1 Purpose

After doing this lab exercise, students are able : 1. Students are able to make bruteforce and divide-conquer algorithms 2. Students are able to apply the use of bruteforce and divide-conquer algorithms

4.2 Calculate Factorial with Brute Force and Divide and Conquer Algorithm

look at the following class diagram:

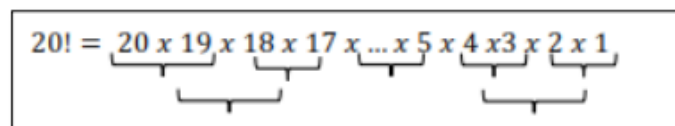| Factorial |
|---|
| number: int |
| factorialBF(int n): int |
| factorialDC(int n): int |
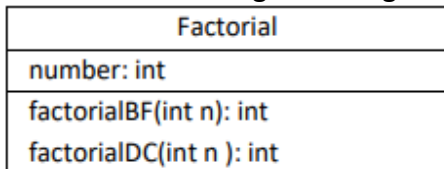
Based on the class diagram above, a Java class program will be created. To calculate the factorial value of a number using 2 types of algorithms, Brute Force and Divide and Conquer. There are differences in the calculation process of the 2 types of algorithms as follows:

Factorial using Brute Force Algorithm:

$$20! = 20 \times 19 \times 18 \times 17 \times \ldots \times 5 \times 4 \times 3 \times 2 \times 1$$
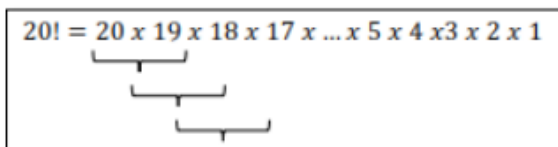
Factorial using Divide and Conquer Algorithm:

$$20! = 20 \times 19 \times 18 \times 17 \times \ldots \times 5 \times 4 \times 3 \times 2 \times 1$$

4.2.1 Lab Unit - 1

1. Create new project "BruteForceDivideConquer". Create new package "week5".
2. Create new class with the name "Factorial00", replace 00 with your absent number
3. Inside "Factorial00" class, add attribute and method according to the class diagram above.
   here is the program code to add attributes and methods:
   a) Add attribute with name number

   ```
   int number;
   ```

   b) add factorialBF() method

   ```java
   public int factorialBF(int n){
       int factorial =1;
       for (int i=1 ;i<=n ;i++){
           factorial=factorial*i;
       }

       return factorial;
   }
   ```

   c) Add faktorialDC() method

```java
public int factorialDC(int n){
        if(n==1){
            return 1;
        }
        else{
            int factorial = n *factorialDC(n-1);
            return factorial;
        }
    }
```

4. Try Running your program (Run) "Factorial00" class by creating a new class with name "FactorialMain00". replace 00 with your absent number
    a) Create Main Method inside FactorialMain00 class.
    b) Inside the main method provide communication with the user to input how many elements will be calculated

```java
Scanner sc00 = new Scanner(System.in);

System.out.println("===========================");
System.out.print("input number of elements : ");
int numElement = sc00.nextInt();
```

    c) Create an Array of Objects in the main function, then input some numbers to calculate the factorial result

```java
factorial00 fk[] = new factorial00[numElement];

for (int i=0;i<fk.length;i++){
    fk[i]=new factorial00();
    System.out.print("input number "+(i+1)+" : ");
    fk[i].number=sc00.nextInt();
}
```

    d) Call faktorialDC() and faktorialBF()method and display the result

```java
System.out.println("===========================");
System.out.println("Factorial result using Brute Force");
for (int i=0;i<fk.length;i++){
    System.out.println("factorial of "+fk[i].number+" is: "+fk[i].factorialBF(fk[i].number));
}
System.out.println("\n===========================");
System.out.println("Factorial result using Divide Conquer");
for (int i=0;i<fk.length;i++){
    System.out.println("factorial of "+fk[i].number+" is: "+fk[i].factorialDC(fk[i].number));
}
System.out.println("\n===========================");
```

    e) Make sure the program is running well!!

4.2.2 Verification
Match the results of your compiled program code

Answer program Lab Unit - 1

```java
src > week5 > factorial08.java > ...
1    package week5;
2
3    public class factorial08 {
4        int number;
5
6        public int factorialBF(int n){
7            int factorial = 1;
8            for (int i=1; i<=n; i++){
9                factorial=factorial*i;
10           }
11           return factorial;
12       }
13
14       public int factorialDC(int n){
15           if(n==1){
16               return 1;
17           }
18           else{
19               int factorial = n*factorialDC(n-1);
20               return factorial;
21           }
22       }
23   }
```

```java
src > week5 > factorialMain08.java > factorialMain08 > main(String[])
1    package week5;
2    import java.util.Scanner;
3
4    public class factorialMain08 {
         Run | Debug
5        public static void main(String[] args) {
6
7            Scanner sc08 = new Scanner(System.in);
8
9            System.out.println("===============================");
10           System.out.println("input number of elements : ");
11           int numElement = sc08.nextInt();
12
13           factorial08 fk[] = new factorial08[numElement];
14
15           for (int i=0; i<fk.length; i++){
16               fk[i]=new factorial08();
17               System.out.println("input number " + (i+1) + " : ");
18               fk[i].number=sc08.nextInt();
19           }
20
21           System.out.println("===============================");
22           System.out.println("Factorial result using Brute Force");
23           for (int i=0; i<fk.length; i++){
24               System.out.println("Factorial of "+fk[i].number+" is : "+fk[i].factorialBF(fk[i].number));
25           }
26           System.out.println("===============================");
27           System.out.println("Factorial result using Divide Conquer");
28           for (int i=0; i<fk.length; i++){
29               System.out.println("Factorial of "+fk[i].number+" is : "+fk[i].factorialDC(fk[i].number));
30           }
31           System.out.println("===============================");
32       }
33   }
```

The result of running the program Lab Unit – 1

```
input number of elements : 3
input number 1 : 5
input number 2 : 8
input number 3 : 3
==========================
Factorial result using Brute Force
factorial of 5 is: 120
factorial of 8 is: 40320
factorial of 3 is: 6

==========================
Factorial result using Divide Conquer
factorial of 5 is: 120
factorial of 8 is: 40320
factorial of 3 is: 6
```

4.2.3 Question

1. Explain the best case of the Divide Conquer Algorithm to calculate factorial values!
   Answer: divide the problem into several parts then solve one by one and combine the results at the end

2. In the implementation of the Divide and Conquer Algorithm in the Factorial case. Does it consist of 3 stages: divide, conquer, and combine? Explain divide, conquer, and combine in the program code above!
   Answer:
   - division stage: in class [factorial 08]
   - conquest stage: on the upper [factorial Main 08] class
   - combine stage: in the lower [factorialMain08] class

3. Is it possible to change the iteration of the factorialBF() method using anything other than the "for" syntax? prove it with program code!
   Answer: no because it is useful for looping

4. Add execution time for both method types and display execution time result!
   Answer:
   - BF execution time:
     ```
     public int factorialBF(int n){
     ```
   - BF execution result:
     ```
     System.out.println("Factorial of "+fk[i].number+" is : "+fk[i].factorialBF(fk[i].number));
     ```
   - DC execution time:
     ```
     public int factorialDC(int n)
     ```
   - DC execution result:
     ```
     System.out.println("Factorial of "+fk[i].number+" is : "+fk[i].factorialDC(fk[i].number));
     ```

5. Prove it by inputting the number of elements above 20, is there any difference in execution time?
   Answer: yes, there is a difference in the results

4.3 Calculate Power of Number using Brute Force and Divide – Conquer Algorithm
In this practicum, we will create a class program in Java. To calculate the power of a number using 2 types of algorithms, Brute Force and Divide and Conquer.

4.3.1 Lab – unit 2

1. Inside week5 package, create a new class with the name Power00. replace 00 with your absent number. Inside Power00 class, create attribute with the name base and exponent. this attribute will be used to calculate Power of number.

```
public class Power00 {
    int base, exponent;

}
```

2. Inside Power00 class, add PowerBF()method

```
public int powerBF(int b, int e){
    int result=1;
    for(int i = 0; i<e; i++){
        result=result*b;
    }
    return result;
}
```

3. Inside Power00 class, add PowerDC()method

```
public int powerDC(int b, int e){
    if(e==0){
        return 1;
    }
    else{
        if(e%2==1){//for odd number
            return (powerDC(b, e/2)*powerDC(b, e/2)*b);
        }
        else{
            return (powerDC(b, e/2)*powerDC(b, e/2));
        }
    }
}
```

4. See if there are no errors that appear in the creation of the Power00 class.
5. Create new class with the name PowerMain00 Class. replace 00 with your absent number. Add main method Inside Power00 class. Add the following code in the main class to input the number of elements to be calculated.

```
Scanner sc00 = new Scanner(System.in);

System.out.println("===========================");
System.out.print("input number of elements : ");
int numElement = sc00.nextInt();
```

6. Element variable in the step 5 will be used to declare and instantiation array of objek. The program code below is for base and exponential input, so if it is written in mathematical form: baseexponent.

```
Power00 pw[]= new Power00[numElement];

for(int i=0; i<pw.length;i++){
    pw[i]=new Power00();
    System.out.print("input base number "+(i+1)+" : ");
    pw[i].base=sc00.nextInt();
    System.out.print("input exponent number "+(i+1)+" : ");
    pw[i].exponent=sc00.nextInt();
}
```

7. Call powerDC() and powerBF()method and display the result

```
System.out.println("=========================");
System.out.println("Power result using Brute Force");
for (int i=0;i<pw.length;i++){
    System.out.println("Power of "+pw[i].base+"^"+pw[i].exponent+" is: "+pw[i].powerBF(pw[i].base, pw[i].exponent));
}
System.out.println("\n=========================");
System.out.println("Power result using Divide Conquer");
for (int i=0;i<pw.length;i++){
    System.out.println("Power of "+pw[i].base+"^"+pw[i].exponent+" is: "+pw[i].powerDC(pw[i].base, pw[i].exponent));
}
System.out.println("\n=========================");
```

4.3.2 Verification
Match the results of your compiled program code.

Answer program Lab Unit - 2

```
src > week5 >  power08.java > ...
1    package week5;
2
3    public class power08 {
4        int base, exponent;
5
6        public int powerBF(int b, int e){
7            int result=1;
8            for(int i=0; i<e; i++){
9                result=result*b;
10           }
11           return result;
12       }
13
14       public int powerDC(int b, int e){
15           if(e==0){
16               return 1;
17           }
18           else{
19               if(e%2==1){
20                   return (powerDC(b, e/2)*powerDC(b, e/2)*b);
21               }
22               else{
23                   return (powerDC(b, e/2)*powerDC(b, e/2));
24               }
25           }
26       }
27   }
```

```
src > week5 > ● powerMain08.java > ✤ powerMain08 > ۞ main(String[])
  1   package week5;
  2   import java.util.Scanner;
  3   public class powerMain08 {
        Run | Debug
  4       public static void main(String[] args) {
  5           Scanner sc08 = new Scanner(System.in);
  6
  7           System.out.println("=============================");
  8           System.out.println("input number of elements : ");
  9           int numElement = sc08.nextInt();
 10
 11           power08 pw[] = new power08[numElement];
 12
 13           for(int i=0; i<pw.length; i++){
 14               pw[i]=new power08();
 15               System.out.println("input base number " + (i+1) + " : ");
 16               pw[i].base=sc08.nextInt();
 17               System.out.println("input exponent number " + (i+1) + " : ");
 18               pw[i].exponent=sc08.nextInt();
 19           }
 20
 21           System.out.println("=============================");
 22           System.out.println("Power result using Brute Force");
 23           for (int i=0; i<pw.length; i++){
 24               System.out.println("Power of "+pw[i].base+" is : "+pw[i].powerBF(pw[i].base, pw[i].exponent));
 25           }
 26           System.out.println("=============================");
 27           System.out.println("Power result using Divide Conquer");
 28           for (int i=0; i<pw.length; i++){
 29               System.out.println("Power of "+pw[i].base+" is : "+pw[i].powerBF(pw[i].base, pw[i].exponent));
 30           }
 31           System.out.println("=============================");
 32       }
 33
 34   }
```

The result of running the program Lab Unit - 2

```
=============================
input number of elements : 2
input base number 1 : 6
input exponent number 1 : 2
input base number 2 : 4
input exponent number 2 : 3
=============================
Power result using Brute Force
Power of 6^2 is: 36
Power of 4^3 is: 64


=============================
Power result using Divide Conquer
Power of 6^2 is: 36
Power of 4^3 is: 64

=============================
```

4.3.3 Question

1. Explain the differences between the 2 methods, PangkatBF() and PangkatDC()!
   Answer:
   - In powerBD : FOR is used, for looping

- In powerDC : we use IF ELSE and recursive functions

2. In the PangkatDC() method there is a snippet of the program as follows:

```
if(e%2==1){//for odd number
    return (powerDC(b, e/2)*powerDC(b, e/2)*b);
}
else{
    return (powerDC(b, e/2)*powerDC(b, e/2));
}
```

Explain the meaning of the code!

Answer: the code to divide the main problem into parts

3. Is there a combine stage in the code? Show it!

Answer:

```
•  return (powerDC(b, e/2)*powerDC(b, e/2)*b);
•  return (powerDC(b, e/2)*powerDC(b, e/2));
```

4. Modify the program code, assume the process of assigning values to attributes is done with a constructor.

5. Add a menu so that only one of the selected methods will be executed!

4.4 Calculate Sum of Array using Brute Force and Divide - Conquer Algorithm

In this experiment, we will practice how the divide, conquer, and combine process is applied to a case study of adding up a company's profits in a few months.

4.4.1 Lab-unit 3

1. Inside week5 pakage Create new class with name Sum00. replace 00 with your absent number. Add attribute inside Sum00 class, the attribute is number of elemets, profit array. Also add a constructor to Sum00 class.

```
public class Sum00 {
    public int element;
    public double profit[];


    public Sum00(int elements){
        this.element=elements;
        this.profit=new double[elements];


    }
}
```

2. Add TotalBF()method which will calculate the total profit using Brute Force

```
double totalBF(double arr[]){
    double totalProfit=0;
    for(int i=0; i<arr.length;i++){
        totalProfit=totalProfit+arr[i];
    }
    return totalProfit;
}
```

3. Add TotalDC()which will calculate the total profit using Divide and Conquer

```
double totalDC(double arr[], int l, int r){
    if(l==r){
        return arr[l];
    }
    else if(l<r){
        int mid = (l+r)/2;
        double lsum = totalDC(arr, l, mid-1);
        double rsum = totalDC(arr, mid+1, r);
        return lsum+rsum+arr[mid];
    }
    return 0;
}
```

4. Create new class MainSum00. replace 00 with your absent number. Add main method inside MainSum00. In this method the user can write down how many months the profit will be calculated. In this class, instantiation of objects is created to call attributes or functions in the Sum class

```
System.out.println("=================================================");
System.out.println("Program to calculate profit of company (units in million, ex: 8 millions");
System.out.print("input number of months : ");
int numMonth = sc00.nextInt();
```

5. Because what will be calculated is the total profit, then also added to the main method, the array will be calculated. The array is an attribute contained in the Sum class, therefore it is necessary to create a Sum object first.

```
Sum00 sm=new Sum00(numMonth);
System.out.println("=================================================");
for(int i=0;i<numMonth;i++){
    System.out.print("input profit for month "+(i+1)+" : ");
    sm.profit[i]=sc00.nextDouble();
}
```

6. Display calculation results through objects that have been created for both ways (Brute Force and Divide and Conquer)

```
System.out.println("=================================================");
System.out.println("Sum of profit using Brute force");
System.out.println("sum of profit : "+sm.totalBF(sm.profit));

System.out.println("=================================================");
System.out.println("Sum of profit using Divide Conquer");
System.out.println("sum of profit : "+sm.totalDC(sm.profit,0,numMonth-1));
```

4.4.2 Verification

Match the results of your compiled program code.

Answer program Lab Unit – 3

```java
src > week5 > sum08.java > sum08
1    package week5;
2
3    public class sum08 {
4        public int element;
5        public double profit[];
6
7        public sum08(int elements){
8            this.element=elements;
9            this.profit=new double[elements];
10       }
11
12       double totalBF(double arr[]){
13           double totalProfit=0;
14           for(int i=0; i<arr.length; i++){
15               totalProfit=totalProfit+arr[i];
16           }
17           return totalProfit;
18       }
19
20       double totalDC(double arr[], int l, int r){
21           if(l==r){
22               return arr[l];
23           }
24           else if(l<r){
25               int mid = (l+r){
26                   double lsum = totalDC(arr, l, mid-1);
27                   double rsum = totalDC(arr, mid=1, r);
28                   return lsum+rsum+arr[mid];
29               }
30               return 0;
31           }
32       }
33   }
34
```

```java
src > week5 > mainSum08.java > ...
1    package week5;
2    import java.util.Scanner;
3
4    public class mainSum08 {
         Run | Debug
5        public static void main(String[] args) {
6            Scanner sc08 = new Scanner(System.in);
7            System.out.println("==============================");
8            System.out.println("Program to calculate profit of company (units in million, ex:8 milllions");
9            System.out.println("input number of months : ");
10           int numMonth = sc08.nextInt();
11
12           sum08 sm=new sum08(numMonth);
13           System.out.println("==============================");
14           for(int i=0; i<numMonth; i++){
15               System.out.println("input profit for month " +(i=1)+ " : ");
16               sm.profit[i]=sc08.nextDouble();
17           }
18
19           System.out.println("==============================");
20           System.out.println("Sum of profit using Brute force");
21           System.out.println("sum of profit : " + sm.totalBF(sm.profit));
22
23           System.out.println("==============================");
24           System.out.println("Sum of profit using Divide Conquer");
25           System.out.println("sum of profit : " + sm.totalDC(sm.profit,0,numMonth-1));
26       }
27   }
28
```

The result of running the program Lab Unit - 3

```
Program to calculate profit of company (units in million, ex: 8 millions
input number of months : 5
=============================================
input profit for month 1 : 8.5
input profit for month 2 : 9.54
input profit for month 3 : 7.2
input profit for month 4 : 9.1
input profit for month 5 : 6
=============================================
Sum of profit using Brute force
sum of profit : 40.339999999999996
=============================================
Sum of profit using Divide Conquer
sum of profit : 40.34
```

4.4.3 Question

1. Give an illustration of the difference in calculating profits between TotalBF() and TotalDC() methods
   Answer:
   - TotalBF() function : use for loop
   - TotalBF() advantage :user enters array inputted in parameter
   - TotalDC() function : use if else for recursive function
   - TotalDC() advantage : divide the problem then solve by elimination process with if else then combine the results at the end

2. Note that the output of the two types of algorithms may have different results after commas. How to delimit the output after commas so that it is standard for both types of algorithms. For example : 40.34
   Answer:
   - use the import java.util.* format; (more simple)
   - use a decimal format like import java.text.DecimalFormat;

3. Why is there the following return value formulation? Explain!

   ```
   return lsum+rsum+arr[mid];
   ```

   Answer: returns the value stored in the variable into the code that will call the function

4. Why is the mid variable needed in the TotalDC() method?
   Answer: make correct and appropriate calculations

5. This program for calculating a company's profits is only for one company. How to calculate the profit of several months at once for several companies.(Each company may have a different number of months)? Prove it with the program!
   Answer: by creating BF and DC containing the length of the month, then inputting each month, and adding them up at the end of the program

4.5 Exercise

A university in the city of Malang is holding a vote to elect the chairman of the BEM in 2022. If the number of votes collected is assumed to be even. So with the input of the selected candidates, find the majority of the votes for each candidate. (The number of array elements and the result of vote selection are user input). Majority Element : The majority element in A is the element which is in

more than n/2 positions. For example, if n=6 or n=7 then the majority value is at least 4. Derived from (7/2)+1 or (6/2)+1).

The majority value is different from the concept by calculating the total number of votes for the chosen candidate!

Answer program Exercise

```java
src > week5 > exercise08.java > exercise08 > main(String[])
1    package week5;
2    import java.util.Scanner;
3
4    public class exercise08 {
         Run | Debug
5        public static void main(String[] args){
6
7            Scanner sc08 = new Scanner(System.in);
8
9            String[] name = new String[50];
10           int[] count = new int[50];
11           int total,constituency,choice;
12
13           System.out.print("Enter total Candidates: ");
14           total=sc08.nextInt();
15           System.out.print("Enter the total of the Voters: ");
16           constituency=sc08.nextInt();
17           System.out.println("======================");
18           sc08.nextLine();
19
20           for(int i=0;i<total;i++) {
21               System.out.print("Candidate num-"+(i+1)+" :");
22               name[i]=sc08.nextLine();
23           }
24
25           System.out.println("Welcome Voters!\nUse your right to vote!");
26           System.out.println("======================");
27           System.out.print("The following is a list of candidates that can be selected: ");
28
29           for(int j=0;j<total;j++) {
30               System.out.print("\n"+(j+1)+". "+name[j]);
31           }
32           System.out.print("\nNote: Enter number!\n");
33           System.out.println("======================");
34           for(int j=0;j<constituency;j++) {
35               System.out.print("Voter num-"+(j+1)+", Choose : ");
36               choice=sc08.nextInt();
37               do {
38                   count[choice] = count[choice] + 1;
39               }while(choice>total);
```

```
43          System.out.println("======================");
44          System.out.println("Elected Chair");
45          for(int k=1;k<=total;k++) {
46              if(count[k]>=((constituency/2)+1)) {
47                  System.out.println("Mayority = "+name[k-1]+", With total votes = "+count[k]);
48              }
49          }
50          System.out.print("Minimum Majority Score = "+((constituency/2)+1));
51      }
52  }
```

The result of running the program Exercise

```
Enter total Candidates: 2
Enter the total of the Voters: 4
======================
Candidate num-1 :Clea
Candidate num-2 :Cleo
Welcome Voters!
Use your right to vote!
======================
The following is a list of candidates that can be selected:
1. Clea
2. Cleo
Note: Enter number!
======================
Voter num-1, Choose : 1
Voter num-2, Choose : 1
Voter num-3, Choose : 1
Voter num-4, Choose : 2
======================
Elected Chair
Mayority = Clea, With total votes = 3
Minimum Majority Score = 3
```