Chapter 4

RECURSIVE

Objective

- To introduce:

    Recursive concept

    Recursive operations

    Recursive implementation

CONTENT

4.1 Introduction

4.2 Implementing recursion approaches

4.2.1 Factorial

4.2.2 Fibonacci

4.1 Introduction

- Is a repetitive process – running the same process continuously (an algorithm that calls itself).

- Recursive:
  - ✓ Procedure, function or method that calls itself.
  - ✓ Allows the same process to be executed continuously using different parameters.
  - ✓ Function value will be returned in a reverse order.

- Advantages:
  - ✓ Algorithm analysis is pretty easy.
  - ✓ Coding/program verification is much easier.

4.1 Introduction

- Example:

Given two positive integers m and n, where m <= n, find

$$jumlahKuasaDua(m, n) = m^2 + (m+1)^2 + \ldots + n^2.$$

e.g.: Assume m = 5; n = 10; and we are trying to find the total of these number from m to n (power of 2).

$$jumlahKuasaDua(5, 10) = 5^2 + 6^2 + 7^2 + 8^2 + 9^2 + 10^2 = 355$$

4.1 Introduction

- Solution without recursive approach:

```
int jumlahKuasadua(int m, int n)
{
    int i, sum;
    sum = 0;

    for (i = m, i <= n, i++)
        sum += i * i;
  return sum;
}
```

4.1 Introduction

- Solution using recursive approach:

  Algorithm:

1. **IF** there is more than one number in the range m:n & m<n

   1.1 Solution is by adding m^2 to the total of square in the range of m+1:n and return the final value.

2. **ELSE**

   2.1 There is only one number in the range of m:n, where m = n, and the solution is m^2; return the final value.

3. **END**

4.1 Introduction

```
int jumlahKuasaDua (int m, int n)
{
    if (m < n)
        return m*m + jumlahKuasaDua(m+1, n);
    else
        return m*m;
}
```

4.1 Introduction

- Traversing the function of jumlahKuasaDua (5,10):

= (25 + jumlahKuasaDua(6,10))

= (25 + (36 + jumlahKuasaDua(7,10)))

= (25 + (36 + (49 + (jumlahKuasaDua(8,10))))

= (25 + (36 + (49 + (64 + jumlahKuasaDua (9,10)))))

= (25 + (36 + (49 + (64 + (81 + jumlahKuasaDua(10,10))))))

= (25 + (36 + (49 + (64 + (81 + 100)))))

= (25 + (36 + (49 + (64 + 181))))

= (25 + (36 + (49 + 245)))

= (25 + (36 + 294))

= (25 + 330)

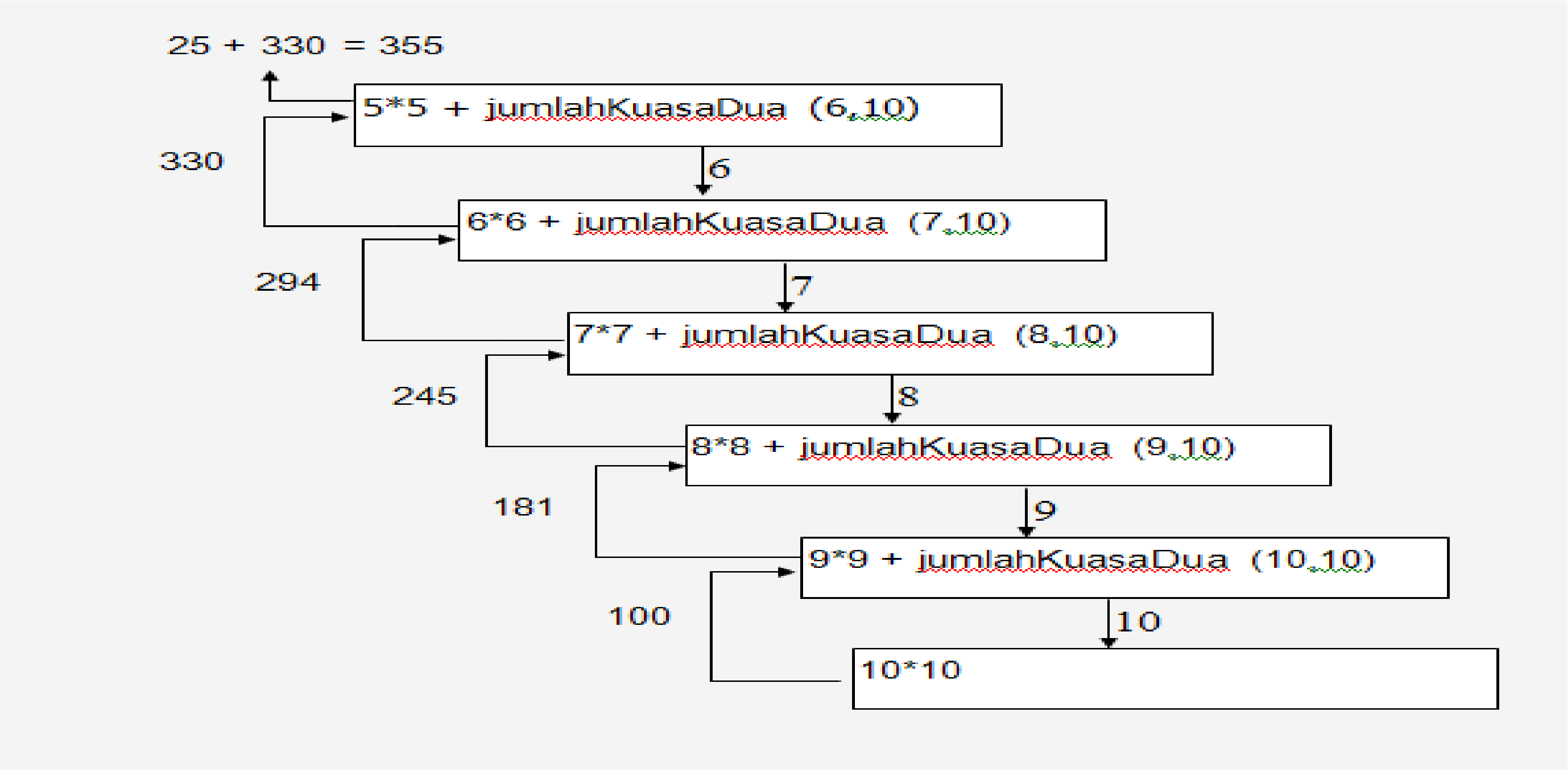
= 355

4.1 Introduction

Diagram:

Using the same problem:

jumlahKuasaDua (5, 10)



4.2 IMPLEMENTING RECURSION APPROACH

FACTORIAL

- Given a factorial n! for integer n (non-negative).

  $n! = 1 \times 2 \times \ldots \times n$ for $n > 0$, and

  $0! = 1$

therefore,

$0! = 1$

$1! = 1$

$2! = 1 \times 2 = 2$

$3! = 1 \times 2 \times 3 = 2! \times 3 = 6$

$4! = 1 \times 2 \times 3 \times 4 = 3! \times 4 = 24$

$5! = 1 \times 2 \times 3 \times 4 \times 5 = 4! \times 5 = 120$

$$\boxed{n! = n \times (n-1)!}$$

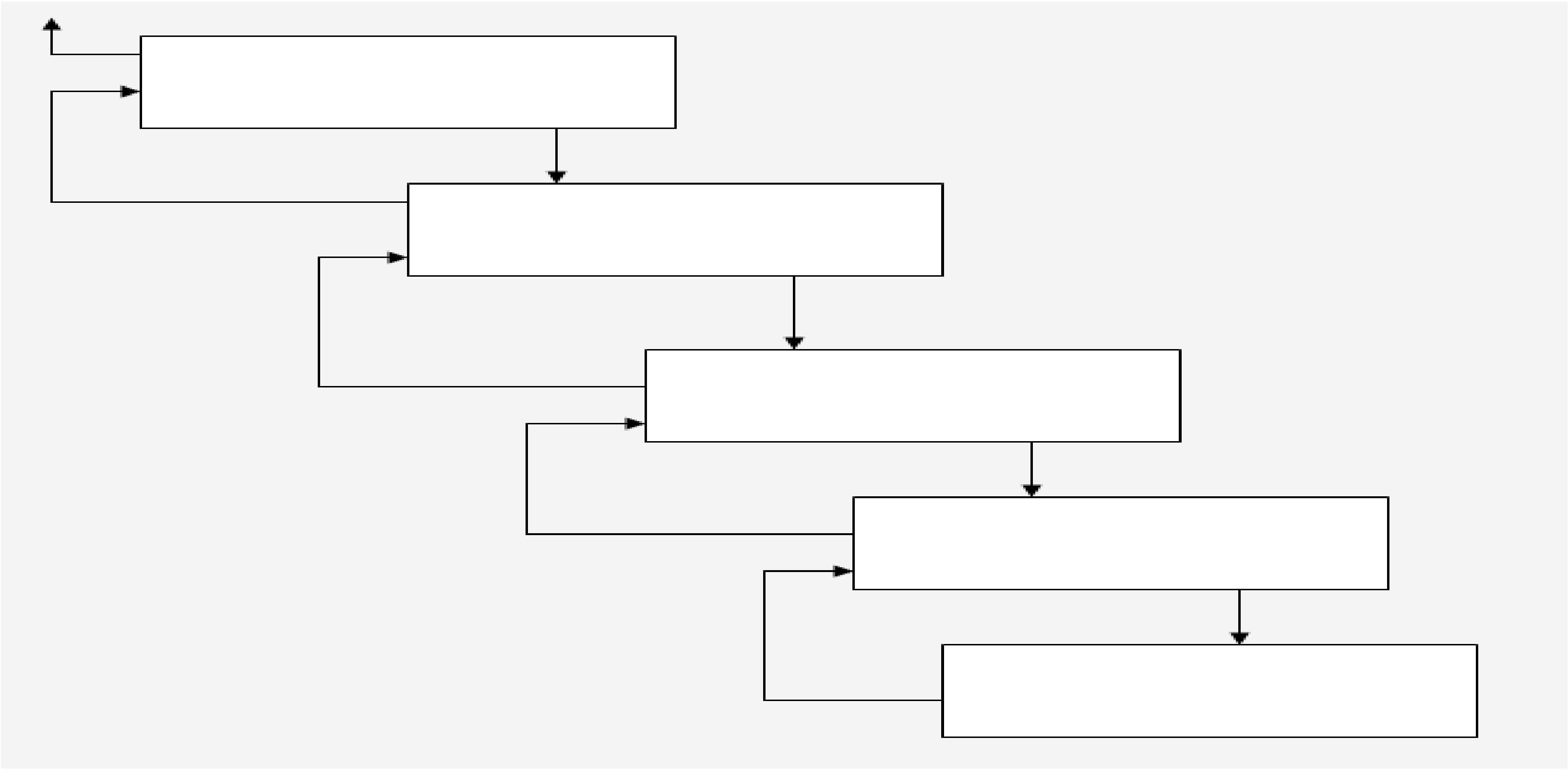
4.2 IMPLEMENTING RECURSION APPROACH

- Code:

```
double Fact (int n)
{
        if (n == 0)
            return 1;
        else
            return Fact (n - 1) * n;
}
```

4.2 IMPLEMENTING RECURSION APPROACH

- Recursive diagram:
  Factorial = Fact (4)



4.2 IMPLEMENTING RECURSION APPROACH

FIBONACCI NUMBER

- Following are the first 10 Fibonacci numbers:

  0, 1, 1, 2, 3, 5, 8, 13, 21, 34, …

  starting with two Fibonacci number (0 and 1), and followed by the other Fibonacci numbers (the sum of 2 previous numbers)

- Therefore, we can define Fibonacci numbers as

$F_0 = 0$

$F_1 = 1$

$F_n = F_{n-1} + F_{n-2}; \text{ where } n > 1$

F_n presenting the n^{th}-number that is in the sequence.

4.2 IMPLEMENTING RECURSION APPROACH

- Code:

```
double Fib (int n)
{
     if (n = = 0)
               return 0;
    else if (n < = 2)
               return 1;
    else
               return Fib (n-1) + Fib(n-2);
}
```

4.2 IMPLEMENTING RECURSION APPROACH

- Recursion diagram:

Fibonacci = Fib(5)

Exercise

1. Based on the recursive method below, draw the recursive diagram and determined what is the output of **abc** method if x = 3 and n = 5?

```
double abc (double x, int n)
{
    if (n == 0)
        return 1.0;
    else
        return abc(x, n – 1) * x;
}
```

Exercise

2. Write a recursive method **power2** to compute x^n using the following recursive formulation:

$$x^0 = 1$$

$$x^n = x * x^{n-1} \text{ if } n > 0$$