**KIST COLLEGE OF MANAGEMENT**

(Affiliated to Tribhuvan university)



**Project Report**

Of

**“Artificial Intelligence”**

**Under Supervision of**

Mr. Tej Bahadur Shahi

**Submitted By**

Anjan Pudasaini (7210/16)

**Submitted To**

**TRIBHUVAN UNIVERSITY**

Faculty of management

Kirtipur, Kathmandu, Nepal

March, 2021

**Table of Contents**

[Lab 1 3](#_Toc65180194)

[Relationship Programs 3](#_Toc65180195)

[Lab 2 4](#_Toc65180196)

[Recursive Programs 4](#_Toc65180197)

[Factorial: 4](#_Toc65180198)

[Fibonacci Series: 5](#_Toc65180199)

[Lab 3 6](#_Toc65180200)

[Tower of Hanoi (TOH) 6](#_Toc65180201)

[Lab 4 8](#_Toc65180202)

[Monkey Banana Problem 8](#_Toc65180203)

[Lab 5 10](#_Toc65180204)

[Realization of logic gates 10](#_Toc65180205)

[AND GATE 10](#_Toc65180206)

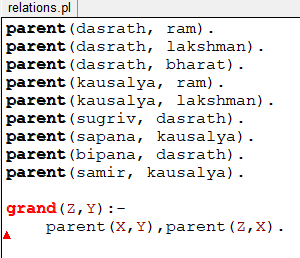
[OR Gate 11](#_Toc65180207)

[NOT Gate 12](#_Toc65180208)

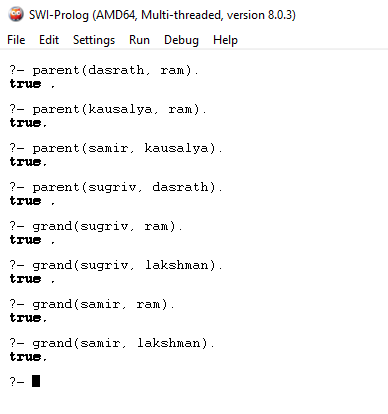
# Lab 1

## Relationship Programs

**Relationship Facts and Source Code:**



**Output:**



# Lab 2

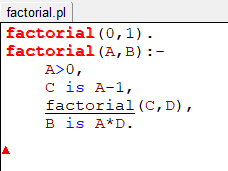
## Recursive Programs

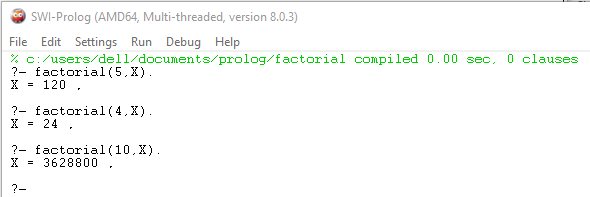
### Factorial:

Factorial of n is the product of all positive descending integers. Factorial of n is denoted by n! For example: -

* 4! = 4\*3\*2\*1 = 24
* 5! = 5\*4\*3\*2\*1 = 120

**Source Code:**

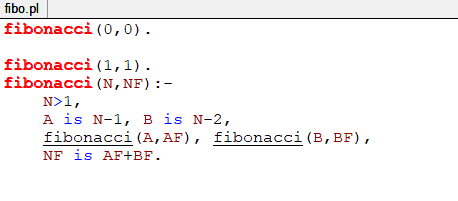


**Output:**

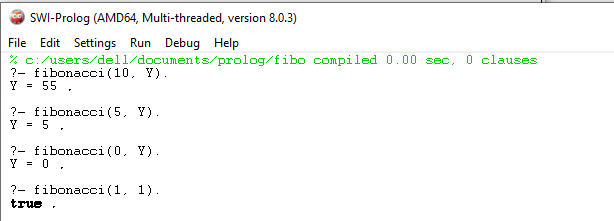
### Fibonacci Series:

In Fibonacci series, next number is the sum of previous two numbers for example 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55 etc. The first two numbers of Fibonacci series are 0 and 1.

**Source Code:**



**Output:**



# Lab 3

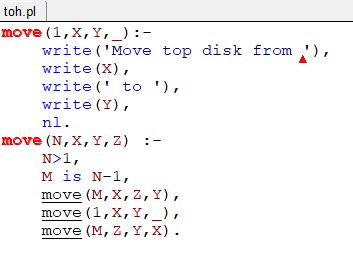
## Tower of Hanoi (TOH)

The Tower of Hanoi is a mathematical game or puzzle. It consists of three rods and a number of disks of different sizes, which can slide onto any rod. The puzzle starts with the disks in a neat stack in ascending order of size on one rod, the smallest at the top, thus making a conical shape.

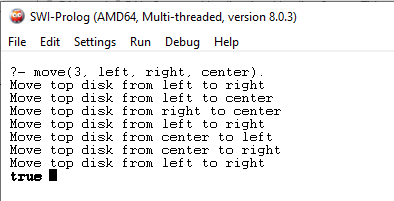
The objective of the puzzle is to move the entire stack to last rod, obeying the following simple rules:

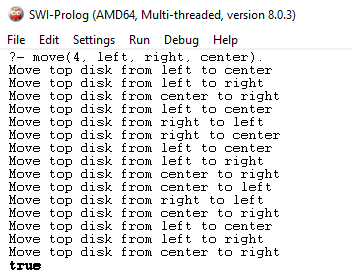
* Only one disk can be moved at a time.
* Each move consists of taking the upper disk from one of the stacks and placing it on top of another stack or on an empty rod.
* No larger disk may be placed on top of a smaller disk.

**Source Code:**



**Output:**





# Lab 4

## Monkey Banana Problem

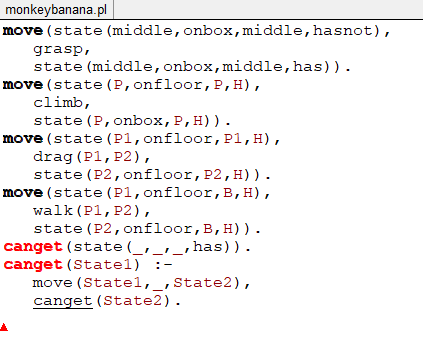
**Problem Statement:**

1. Suppose the problem is as given below −
2. A hungry monkey is in a room, and he is near the door.
3. The monkey is on the floor.
4. Bananas have been hung from the center of the ceiling of the room.
5. There is a block (or chair) present in the room near the window.

**Solution:**

1. Grab the stick
2. Climb on the chair
3. Wave the stick
4. Get off the chair
5. Grab banana

**Source Code:**



**Output:**



# Lab 5

## Realization of logic gates

### AND GATE

The AND gate gives an output of 1 if both the two inputs are 1, it gives 0 otherwise.

**Source Code:**

int main() {

int a[5] = { 1, 0, 1, 0, 1 };

int b[5] = { 0, 1, 1, 0, 0 };

int i, product;

for (i = 0; i < 5; i++) {

// using product method

product = a[i] \* b[i];

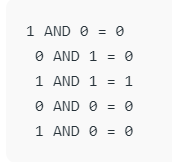
printf("\n %d AND %d = %d",

a[i], b[i], product);

}

}

**Output:**



### OR Gate

The OR gate gives an output of 1 if either of the two inputs are 1, it gives 0 otherwise.

**Source Code:**

int main() {

int a[5] = { 1, 0, 1, 0, 1 };

int b[5] = { 0, 1, 1, 0, 0 };

int i, or\_ans;

for (i = 0; i < 5; i++) {

// using the + operator

if (a[i] + b[i] > 0)

or\_ans = 1;

else

or\_ans = 0;

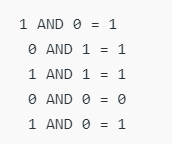
printf("\n %d AND %d = %d",

a[i], b[i], or\_ans);

}

}

**Output**



### NOT Gate

It acts as an inverter. It takes only one input. If the input is given as 1, it will invert the result as 0 and vice-versa.

**Source Code:**

int main() {

int a[5] = { 1, 0, 1, 0, 1 };

int i, ans;

for (i = 0; i < 5; i++) {

if (a[i] == 0)

ans = 1;

else

ans = 0;

printf("\n NOT %d = %d", a[i], ans);

}

}

**Output:**

