EXAMPLE PROGRAM DECIMAL TO BINARY CONVERTER

1. Requirements

Design and Produce some software, written in Ada, that outputs the Nth term in a Fibonacci Sequence:

```
0, 1, 1, 2, 3, 5, 8, 13 ....
```

where the first two, terms are 0 and 1 , and each term thereafter is the sum of the two proceeding terms. Thus:

```
Fib(N) = Fib(N-1) + fib(N-2)
```

where N is greater than 2. Assume that the N is a positive integer in the range 1..30 inclusive.

2. Design

A top down analysis of the problem is given in Table 1.

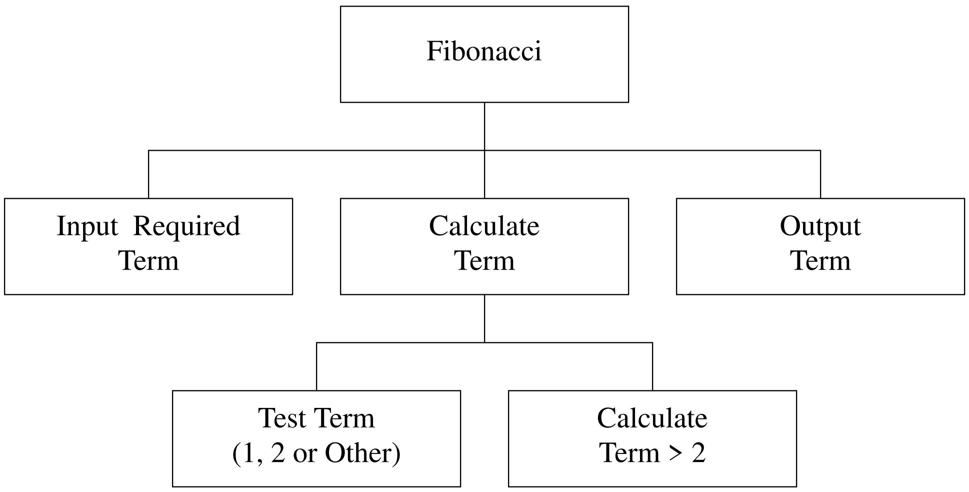


Table 1: *Top down analysis*

The operations identified in Table 1 may be implemented using the following procedures:

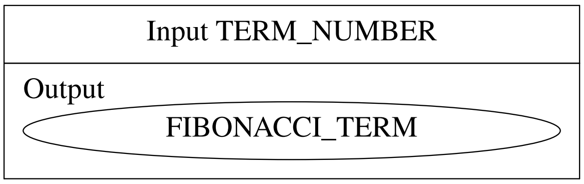
1) `FIBONACCI` (Top level procedure): Input required term, call `FIBONACCI_TERM`, output result.

Name	Description	Type	Range
TERM_NUMBER	Global input variable	POSITIVE	1..30

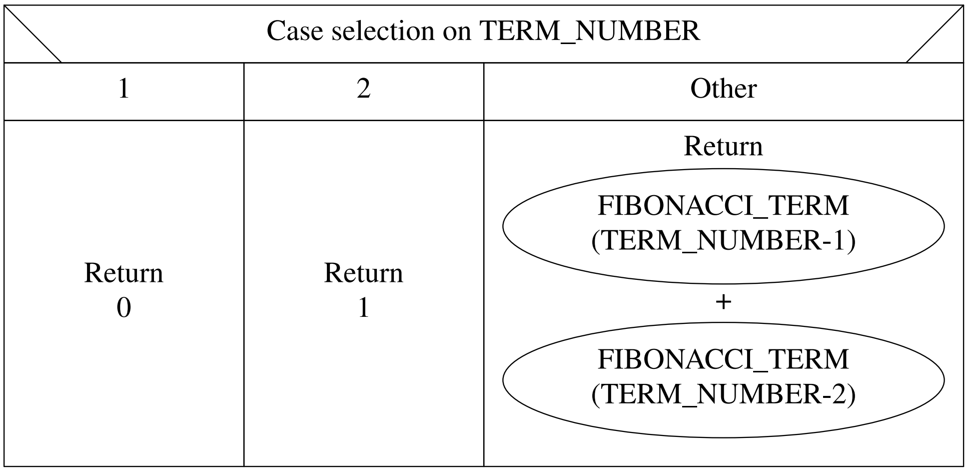
2) `FIBONACCI_TERM` (Level 2 function): Calculate and return term.

Name	Description	Type	Range
TERM_NUM	Formal_parameter	POSITIVE	Default

Appropriate Nassi-Shneiderman design is given in Table 2.



FIBONACCI
(Top level procedure)



FIBONACCI_TERM
(Level 2 function)

Table 2*: Nassi-Schneiderman chart*

3. Implementation

The encoding of the above design is given in Table 3.

```
-- FIBONACCI
-- 17 September 1997
-- Frans Coenen
-- Dept Computer Science, University of Liverpool

with TEXT_IO;
use TEXT_IO;

procedure FIBONACCI is
    package INTEGER_INOUT is new INTEGER_IO(INTEGER);
    use INTEGER_INOUT;
    TERM_NUMBER: POSITIVE range 1..30;

    ----------------------------------------------------------------------
    -- FIBONACCI
    function FIBONACCI_TERM(TERM_NUM: POSITIVE) return INTEGER is
    begin
        case TERM_NUM is
            when 1 =>
                return 0;
            when 2 =>
                return 1;
            when others =>
                return FIBONACCI_TERM(TERM_NUM-1) +
                        FIBONACCI_TERM(TERM_NUM-2);
        end case;
    end FIBONACCI_TERM;
    ----------------------------------------------------------------------

-- TOP LEVEL PROCEDURE
BEGIN
-- Input required term number
    PUT_LINE("Input term  number (positive integer 1..30): ");
    GET(TERM_NUMBER);
-- Calculate and output term
    PUT("Fib(");
    PUT(TERM_NUMBER, WIDTH=>1);
    PUT(") = ");
    PUT(FIBONACCI_TERM(TERM_NUMBER));
    NEW_LINE;
end FIBONACCI;
```

Table 3: *Implementation*

4. Testing

4.1. Black box testing

BVA, Limit and Arithmetic Testing: Test range specification and limits for TERM_NUMBER input variable and arithmetic expressions using test cases of the form presented in Table 4. Results are presented in Table 5.

TEST CASE TERM_NUMBER	EXPECTED RESULT OUTPUT
0	CONSTRAINT_ERROR
1	0
2	1
15	377
29	317811
30	514229
31	CONSTRAINT_ERROR

Table 4: BVA Test Cases

```
kuban-381$ fibonacci
Input term  number (positive integer 1..30):
0
Ada-runtime: Exception CONSTRAINT_ERROR raised in
/tmp_mnt/home/staff2/ra/frans/public_html/2CS21/week8/fibonacci.ada
on line 36. Rangecheck error: 0 not in range 1..30.

kuban-382 $ fibonacci
Input term  number (positive integer 1..30):
1
Fib(1) =           0

kuban-383 $ fibonacci
Input term  number (positive integer 1..30):
2
Fib(2) =           1

kuban-384 $ fibonacci
Input term  number (positive integer 1..30):
15
Fib(15) =         377

kuban-385 $ fibonacci
Input term  number (positive integer 1..30):
29
Fib(29) =      317811

kuban-386 $ fibonacci
Input term  number (positive integer 1..30):
30
Fib(30) =      514229

kuban-387 $ fibonacci
Input term  number (positive integer 1..30):
31
Ada-runtime: Exception CONSTRAINT_ERROR raised in
/tmp_mnt/home/staff2/ra/frans/public_html/2CS21/week8/fibonacci.ada
on line 36. Rangecheck error: 31 not in range 1..30.
```

Table 5: Results of BVA, Limit and arithmetic testing Testing

4.2. White box testing.

Path Testing: Test all paths through FIBONACCI_TERM function. This is adequately achieved by the set of test cases presented in Table 4.

Note: no loop testing will be required.

4.3. Data validation testing

Results of a number of data validation tests are presented in Table 5.

```
kuban-514 $ fibonacci
Input term  number (positive integer 1..30):
1.1
Fib(1) =           0

kuban-515 $ 1
ksh: 1:  not found

kuban-516 $ fibonacci
Input term  number (positive integer 1..30):
frans
Ada-runtime: Exception DATA_ERROR propagated out of main.
kuban-517 $ rans
ksh: rans:  not found
```

Table 5: *Data validation testing results*