EXAMPLE PROGRAM ARITHMETIC PROGRESSION

1. Requirements

Design and implement an Ada program that outputs the first N terms of an a.p. (arithmetic progression) of the form:

A, A+D, A+2D, A+3D ...

where A is the first term and D is the common difference. Assume that A and D are integers within the range -1000..1000, and that N is a natural number whose maximum permissible value is 100.

2. Design

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

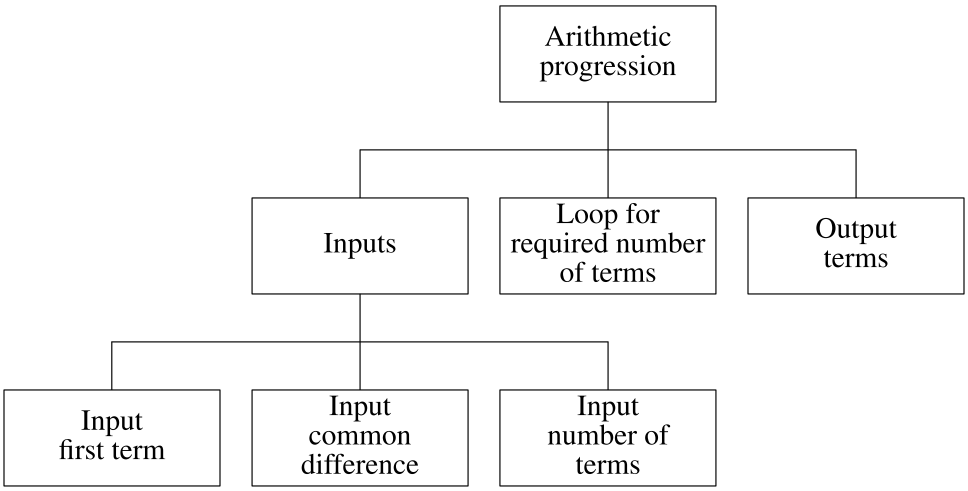

Table 1: *Top down analysis.*

The identified operations can be implemented using two procedures:

1) `ARITHMETIC_PROGRESSION` (Top level procedure): Input data and call to `OUTPUT_TERMS`.

Name	Description	Type	Range
FIRST_TERM	Global input variable	INTEGER	-1000..1000
COMMON_DIFFERENCE	Global input variable	INTEGER	-1000..1000
NUMBER_OF_TERMS	Global input variable	NATURAL	0..100

2) `OUTPUT_TERMS` (level 2 procedure): Fixed count loop to output terms.

Name	Description	Type	Range
FRST_TERM	Formal parameter	INTEGER	Default
COM_DIFF	Formal parameter	INTEGER	Default
NUM_TERMS	Formal parameter	NATURAL	Default
START_CONDITION	Local constant	INTEGER	1

An appropriate Nassi-Shneiderman design is given in Table 2.

Input FIRST_TERM
Input COMMON_DIFFERENCE
Input NUMBER_OF_TERMS
OUTPUT_TERMS(FIRST_TERM, COMMON_DIFFERENCE, NUMBER_OF_TERMS)

ARITHMETIC_PROGRESSION
(Top level procedure)

Fixed count loop: LOOP_PARAM := 1 to NUMBER_OF_TERMS
Output FRST_TERM+(COM_DIFF*LOOP_PARAM)

OUTPUT_TERM
(Level 2 procedure)

Table 2: *Nassi-Shneiderman chart*

3. Implementation

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

```
-- ARITHMETIC PROGRESSION
-- 12 September 1997
-- Frans Coenen
-- Dept Computer Science, University of Liverpool

with TEXT_IO;
use TEXT_IO;

procedure ARITHMETIC_PROGRESSION is
    package INTEGER_INOUT is new INTEGER_IO(INTEGER);
    use INTEGER_INOUT;
    FIRST_TERM, COMMON_DIFFERENCE : INTEGER range -1000..1000;
    NUMBER_OF_TERMS               : NATURAL range 0..100;
```

Table 3: *Implementation*

```
    -------------------------------------------------------------------
    -- OUTPUT TERMS
    procedure OUTPUT_TERMS(FRST_TERM, COM_DIFF: INTEGER; NUM_TERMS:
            NATURAL) is
        START_CONDITION: constant := 1;
    begin
        for LOOP_PARAM in START_CONDITION..NUM_TERMS loop
            PUT(FRST_TERM+(COM_DIFF*LOOP_PARAM));
            PUT(" ");
        end loop;
        NEW_LINE;
    end OUTPUT_TERMS;
    -------------------------------------------------------------------

-- TOP LEVEL
begin
    PUT("Input number of terms (integer number 0..100): ");
    GET(NUMBER_OF_TERMS);
    NEW_LINE;
    PUT("Input first term and common difference ");
    PUT("(integers -1000..1000): ");
    GET(FIRST_TERM);
    GET(COMMON_DIFFERENCE);
    NEW_LINE;
    OUTPUT_TERMS(FIRST_TERM,COMMON_DIFFERENCE,NUMBER_OF_TERMS);
end ARITHMETIC_PROGRESSION;
```

Table 3 continued

4. Testing

4.1. Black box testing

BVA Testing: Suitable set of BVA test cases is given in Table 4 and the results in Table 5.

TEST CASE			RESULT
NUMBER_OF_ TERMS	FIRST_ TERM	COMMON_ DIFFERENCE	OUTPUT
-1	*	*	CONSTRAINT_ERROR
101	*	*	CONSTRAINT_ERROR
1	-1001	*	CONSTRAINT_ERROR
99	1001	*	CONSTRAINT_ERROR
1	-999	-1001	CONSTRAINT_ERROR
99	999	1001	CONSTRAINT_ERROR
1	-999	-999	-999
99	999	999	999, 1998, 2997, 3996 ...

Table 4*: BVA test cases*

```
kuban-274 $ arithmetic_progression
Input number of terms (integer number 0..100): -1
Ada-runtime: Exception CONSTRAINT_ERROR raised in /tmp_mnt/home/
staff2/ra/frans/public_html/2CS21/week7/arithmetic_progression.ada
on line 32. Rangecheck error: -1 not in range 0..100.

kuban-275 $ arithmetic_progression
Input number of terms (integer number 0..100): 101
Ada-runtime: Exception CONSTRAINT_ERROR raised in /tmp_mnt/home/
staff2/ra/frans/public_html/2CS21/week7/arithmetic_progression.ada
on line 32. Rangecheck error: 101 not in range 0..100.

kuban-276 $ arithmetic_progression
Input number of terms (integer number 0..100): 1
Input first term and common difference (integers -1000..1000): -1001
Ada-runtime: Exception CONSTRAINT_ERROR raised in /tmp_mnt/home/
staff2/ra/frans/public_html/2CS21/week7/arithmetic_progression.ada
on line 36. Rangecheck error: -1001 not in range -1000..1000.

kuban-277 $ arithmetic_progression
Input number of terms (integer number 0..100): 99
Input first term and common difference (integers -1000..1000): 1001
Ada-runtime: Exception CONSTRAINT_ERROR raised in /tmp_mnt/home/
staff2/ra/frans/public_html/2CS21/week7/arithmetic_progression.ada
on line 36. Rangecheck error: 1001 not in range -1000..1000.

kuban-278 $ arithmetic_progression
Input number of terms (integer number 0..100): 1
Input first term and common difference (integers -1000..1000): -999  -1001
Ada-runtime: Exception CONSTRAINT_ERROR raised in /tmp_mnt/home/
staff2/ra/frans/public_html/2CS21/week7/arithmetic_progression.ada
on line 37. Rangecheck error: -1001 not in range -1000..1000.

kuban-279 $ arithmetic_progression
Input number of terms (integer number 0..100): 99
Input first term and common difference (integers -1000..1000): 999 1001
Ada-runtime: Exception CONSTRAINT_ERROR raised in /tmp_mnt/home/
staff2/ra/frans/public_html/2CS21/week7/arithmetic_progression.ada
on line 37. Rangecheck error: 1001 not in range -1000..1000.

kuban-280 $ arithmetic_progression
Input number of terms (integer number 0..100): 1
Input first term and common difference (integers -1000..1000): -999 -999
       -999
```

Table 5: *Result of BVA test cases*

```
kuban-294 $ arithmetic_progression
Input number of terms (integer number 0..100): 99
Input first term and common difference (integers -1000..1000): 999 999

  999      1998      2997      3996      4995      5994      6993
 7992      8991      9990     10989     11988     12987     13986
14985     15984     16983     17982     18981     19980     20979
21978     22977     23976     24975     25974     26973     27972
28971     29970     30969     31968     32967     33966     34965
35964     36963     37962     38961     39960     40959     41958
42957     43956     44955     45954     46953     47952     48951
49950     50949     51948     52947     53946     54945     55944
56943     57942     58941     59940     60939     61938     62937
63936     64935     65934     66933     67932     68931     69930
70929     71928     72927     73926     74925     75924     76923
77922     78921     79920     80919     81918     82917     83916
84915     85914     86913     87912     88911     89910     90909
91908     92907     93906     94905     95904     96903     97902
98901
```

Table 5 continued

Limit Testing: Suitable set of limit test cases is given in Table 6 and some sample results in Table 7.

TEST CASE			RESULT
NUMBER_OF_ TERMS	FIRST_ TERM	COMMON_ DIFFERENCE	OUTPUT
100	-1000	-1000	-1000, -2000, -3000, -4000 ...
100	-1000	1000	-1000, 0, 1000, 2000 ...
100	1000	-1000	1000, 0, 1000, 2000 ...
100	1000	1000	1000, -2000, -3000, -4000 ...
0	-1000	-1000	No output
0	-1000	1000	No output
0	1000	-1000	No output
0	1000	1000	No output

Table 6*: Limit test cases*

```
kuban-295 $ arithmetic_progression
Input number of terms (integer number 0..100): 100
Input first term and common difference (integers -1000..1000): -1000 -1000
 -1000      -2000      -3000      -4000      -5000      -6000      -7000
 -8000      -9000     -10000     -11000     -12000     -13000     -14000
-15000     -16000     -17000     -18000     -19000     -20000     -21000
-22000     -23000     -24000     -25000     -26000     -27000     -28000
-29000     -30000     -31000     -32000     -33000     -34000     -35000
-36000     -37000     -38000     -39000     -40000     -41000     -42000
-43000     -44000     -45000     -46000     -47000     -48000     -49000
-50000     -51000     -52000     -53000     -54000     -55000     -56000
-57000     -58000     -59000     -60000     -61000     -62000     -63000
-64000     -65000     -66000     -67000     -68000     -69000     -70000
-71000     -72000     -73000     -74000     -75000     -76000     -77000
-78000     -79000     -80000     -81000     -82000     -83000     -84000
-85000     -86000     -87000     -88000     -89000     -90000     -91000
-92000     -93000     -94000     -95000     -96000     -97000     -98000
-99000    -100000

kuban-296 $ arithmetic_progression
Input number of terms (integer number 0..100): 100
Input first term and common difference (integers -1000..1000): -1000 1000
-1000          0       1000       2000       3000       4000       5000
 6000       7000       8000       9000      10000      11000      12000
13000      14000      15000      16000      17000      18000      19000
20000      21000      22000      23000      24000      25000      26000
27000      28000      29000      30000      31000      32000      33000
34000      35000      36000      37000      38000      39000      40000
41000      42000      43000      44000      45000      46000      47000
48000      49000      50000      51000      52000      53000      54000
55000      56000      57000      58000      59000      60000      61000
62000      63000      64000      65000      66000      67000      68000
69000      70000      71000      72000      73000      74000      75000
76000      77000      78000      79000      80000      81000      82000
83000      84000      85000      86000      87000      88000      89000
90000      91000      92000      93000      94000      95000      96000
97000      98000

kuban-299 $ arithmetic_progression
Input number of terms (integer number 0..100): 0
Input first term and common difference (integers -1000..1000): -1000 -1000

kuban-300 $ arithmetic_progression
Input number of terms (integer number 0..100): 0
Input first term and common difference (integers -1000..1000): -1000 1000
```

Table 7*: Result of limit test cases*

Arithmetic Testing: We should test inputs with negative, zero and positive sample values. This is appropriate for the `FIRST_TERM` and `COMMON DIFFERENCE` inputs. The `NUMBER_OF_TERMS` input cannot have a negative value. We should therefore test this with zero and positive sample values. However, a zero sample value will result in "no passes through the loop", and has been incorporated into loop testing (see below). For the purpose of arithmetic testing we will only use a positive sample value for `NUMBER_OF_TERMS` A suitable set of test cases is given in Table 8 and the results in Table 9.

TEST CASE			RESULT
NUMBER_OF TERMS	FIRST_ TERM	COMMON_ DIFFERENCE	OUTPUT
4	-500	-500	-500, -1000, -1500, -2000
4	-500	0	-500, -500, -500, -500
4	-500	500	-500, 0, 500, 1000
4	0	-500	0, -500, -1000, -1500
4	0	0	0, 0, 0, 0
4	0	500	0, 500, 1000, 1500
4	500	-500	500, 0, -500, -1000
4	500	0	500, 500, 500, 500
4	500	500	500, 1000, 1500, 2000

Table 8: *Arithmetic test cases*

```
kuban-301 $ arithmetic_progression
Input number of terms (integer number 0..100): 4
Input first term and common difference (integers -1000..1000): -500 -500
       -500       -1000       -1500       -2000

kuban-302 $ arithmetic_progression
Input number of terms (integer number 0..100): 4
Input first term and common difference (integers -1000..1000): -500 0
       -500        -500        -500        -500

kuban-303 $ arithmetic_progression
Input number of terms (integer number 0..100): 4
Input first term and common difference (integers -1000..1000): -500 500
       -500           0         500        1000

kuban-304 $ arithmetic_progression
Input number of terms (integer number 0..100): 4
Input first term and common difference (integers -1000..1000): 0 -500
          0        -500       -1000       -1500

kuban-305 $ arithmetic_progression
Input number of terms (integer number 0..100): 4
Input first term and common difference (integers -1000..1000): 0 0
          0           0           0           0

kuban-306 $ arithmetic_progression
Input number of terms (integer number 0..100): 4
Input first term and common difference (integers -1000..1000): 0 500
          0         500        1000        1500

kuban-307 $ arithmetic_progression
Input number of terms (integer number 0..100): 4
Input first term and common difference (integers -1000..1000): 500 -500
        500           0        -500       -1000

kuban-308 $ arithmetic_progression
Input number of terms (integer number 0..100): 4
Input first term and common difference (integers -1000..1000): 500 0
        500         500         500         500
```

Table 9: *Result of Arithmetic test cases*

```
kuban-309 $ arithmetic_progression
Input number of terms (integer number 0..100): 4
Input first term and common difference (integers -1000..1000): 500 500
        500        1000        1500        2000
```

Table 9 Continued

4.2. White box testing

Loop testing:. A set of loop test cases is given in Table 10 (results in Table 11). These are designed to exercise the loop as follows:

- No passes through loop (i.e. skip through loop).
- Only one pass through the loop
- Two passes through the loop
- M passes through the loop where M < END_VALUE
- END_VALUE-1 passes through the loop
- END_VALUE passes through the loop

TEST CASE			RESULT
NUMBER_OF_ TERMS	FIRST_ TERM	COMMON_ DIFFERENCE	OUTPUT
0	4	2	No output
1	4	-2	4
2	0	2	0 2
10	0	0	0 0 0 0 0 0 ...
99	0	-2	0 -2 -4 -6 -8 -10 ...
100	-4	2	-4 -2 0 2 4 6 ..

Table 10: *Loop test cases*

```
kuban-255 $ arithmetic_progression
Input number of terms (integer number 0..100): 0
Input first term and common difference (integers -1000..1000): 4 2
```

Table 11: *Result loop test cases*

```
kuban-256 $ arithmetic_progression
Input number of terms (integer number 0..100): 1
Input first term and common difference (integers): 4 -2
          4

kuban-258 $ arithmetic_progression
Input number of terms (integer number 0..100): 2
Input first term and common difference (integers): 0 2
          0           2

kuban-259 $ arithmetic_progression
Input number of terms (integer number 0..100): 10
Input first term and common difference (integers): 0 0
          0           0           0           0           0           0
          0           0           0           0

kuban-260 $ arithmetic_progression
Input number of terms (integer number 0..100): 99
Input first term and common difference (integers): 0 -2
         0        -2        -4        -6        -8       -10        -1
         2       -14       -16       -18       -20       -22       -24
       -26       -28       -30       -32       -34       -36       -38
       -40       -42       -44       -46       -48       -50       -52
       -54       -56       -58       -60       -62       -64       -66
       -68       -70       -72       -74       -76       -78       -80
       -82       -84       -86       -88       -90       -92       -94
       -96       -98      -100      -102      -104      -106      -108
      -110      -112      -114      -116      -118      -120      -122
      -124      -126      -128      -130      -132      -134      -136
      -138      -140      -142      -144      -146      -148      -150
      -152      -154      -156      -158      -160      -162      -164
      -166      -168      -170      -172      -174      -176      -178
      -180      -182      -184      -186      -188      -190      -192
      -194      -196

kuban-261 $ arithmetic_progression
Input number of terms (integer number 0..100): 100
Input first term and common difference (integers): -4 2
       -4        -2         0         2         4         6        8
       10        12        14        16        18        20       22
       24        26        28        30        32        34       36
       38        40        42        44        46        48       50
       52        54        56        58        60        62       64
       66        68        70        72        74        76       78
       80        82        84        86        88        90       92
       94        96        98       100       102       104      106
      108       110       112       114       116       118      120
      122       124       126       128       130       132      134
      136       138       140       142       144       146      148
      150       152       154       156       158       160      162
      164       166       168       170       172       174      176
      178       180       182       184       186       188      190
      192       194
```

Table 11 Continued

4.3. Data validation testing

Results of some data validation testing is presented in Table 12.

```
kuban-314 $ arithmetic_progression
Input number of terms (integer number 0..100): frans
Ada-runtime: Exception DATA_ERROR propagated out of main.

kuban-315 $ rans
ksh: rans:  not found

kuban-316 $ arithmetic_progression
Input number of terms (integer number 0..100): 2.4
Input first term and common difference (integers -1000..1000):
Ada-runtime: Exception DATA_ERROR propagated out of main.

kuban-317 $ 4
ksh: 4:  not found

kuban-318 $ arithmetic_progression
Input number of terms (integer number 0..100): 10
Input first term and common difference (integers -1000..1000): 2 4.5
          2           6           10           14           18           22
         26          30           34           38

kuban-319 $ 5
ksh: 5:  not found
```

Table 12: *Data validation test case results*