## EXAMPLE PROGRAM LOWER TO UPPER CASE CONVERSION REPORT

### 1. Requirements

To produce a program that converts lower case alphabetic characters to upper case alphabetic characters. Note that lower case letters a..z have ASCII codes 97..122, and upper case letters A..Z have ASCII codes 65..90. Therefore to convert from lower case to upper case we must subtract -32 from the ASCII code of the input character. The ASCII character code of 96 is equivalent to the symbol ’‘’, and a code 123 to the symbol ‘{’.

### 2. Design

A top down analysis is presented in Table 1.

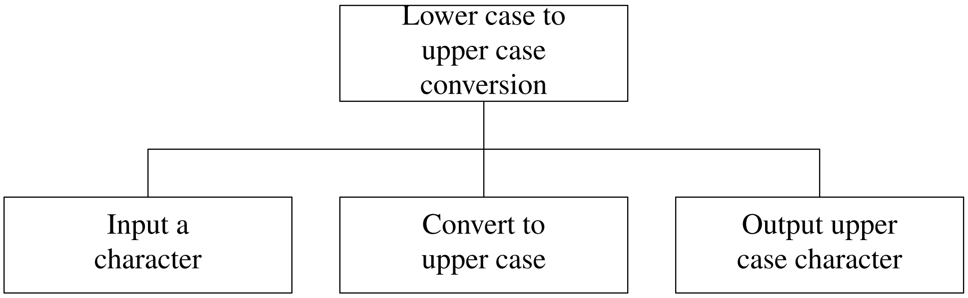


**Table 1:** *Top down analysis*

A single procedure:

1) `LOWER_2_UPPER` (top level procedure) - input lower case character, convert, output upper case character.

| NAME | DESCRIPTION | TYPE | VALUE/RANGE |
|---|---|---|---|
| LOW_CASE_LETTER | Input variable | LOWER_CASE | ’a’..’z’ |
| ASCII_CODE | Global variable | INTEGER | Default |

where `LOWER_CASE` is a subtype of the type `CHARACTER`.

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

| |
|---|
| Input lower case character |
| conversion |
| Output upper case character |

**Table 2:** *Nassi-Shneiderman Chart*

## 3. Implementation

Implementation is as shown in Table 3:

```
-- LOWER TO UPPER CASE CONVERSION
-- 6 August 1997
-- Frans Coenen
-- Dept Computer Science, University of Liverpool

with TEXT_IO;
use TEXT_IO;

procedure LOWER_2_UPPER is
        subtype LOWER_CASE is CHARACTER range 'a'..'z';
        LOW_CASE_LETTER: LOWER_CASE;
        ASCII_CODE: INTEGER;
begin
-- Input lower case character
        PUT_LINE("Input lower case character: ");
        GET(LOW_CASE_LETTER);
-- Conversion
        ASCII_CODE := CHARACTER'POS(LOW_CASE_LETTER)-32;
-- Output upper case value
        PUT("The equivalent in upper case is: ");
        PUT(CHARACTER'VAL(ASCII_CODE));
        NEW_LINE;
end LOWER_2_UPPER;
```

**Table 3:** *Implementation*

## 4. Testing

A set of BVA and limit test cases is given in Table 4 (results presented in Table 5). These test cases will also serve to test the arithmetic operation of the code with the inclusion of a sample input value near the middle of the prescribed range (e.g. `'m'`). We should also carry out some random data validation testing (Table 6).

| TEST CASE | EXPECTED RESULT |
|---|---|
| LOW_CASE_LETTER | Output |
| '`' | CONSTRAINT_ERROR |
| 'a' | 'A' |
| 'b' | 'B' |
| 'm' | 'M' |
| 'y' | 'Y' |
| 'z' | 'Z' |
| '{' | CONSTRAINT_ERROR |

**Table 4:** *BVA, limit and arithmetic test cases*

```
kuban-331 $ lower_2_upper
Input lower case character:
'
Ada-runtime: Exception CONSTRAINT_ERROR raised in /tmp_mnt/home/staff2/
ra/frans/public_html/2CS21/week4/lower_2_upper.ada on line 12.
Rangecheck error: 96 not in range 97..122.

kuban-332 $ lower_2_upper
Input lower case character:
a
The equivalent in upper case is: A

kuban-333 $ lower_2_upper
Input lower case character:
b
The equivalent in upper case is: B

kuban-334 $ lower_2_upper
Input lower case character:
y
The equivalent in upper case is: Y

kuban-335 $ lower_2_upper
Input lower case character:
z
The equivalent in upper case is: Z

kuban-336 $ lower_2_upper
Input lower case character:
{
Ada-runtime: Exception CONSTRAINT_ERROR raised in /tmp_mnt/home/staff2/
ra/frans/public_html/2CS21/week4/lower_2_upper.ada on line 12.
Rangecheck error: 123 not in range 97..122.
```

**Table 5:** *BVA, limit and arithmetic test output*

```
kuban-214 $ lower_2_upper
Input lower case character:
frans
The equivalent in upper case is: F

kuban-215 $ rans
ksh: rans:  not found
```

**Table 6:** *Data validation tests*