Introduction to Computers and Programming

Prof. I. K. Lundqvist

Lecture 2
Mar 12 2004

Concept Question:
What is the Output?

1. 1, 1
2. 1, 49
3. Don’t know

Characters and Strings

- Related Packages
- Operations on both types and the differences between them
- Manipulation of Strings

Character and Wide Character

- **Character**: correspond to the 256 code positions of Row 00 (also known as Latin-1) of the ISO 10646 Basic Multilingual Plane (BMP).

- **Wide_Character**: correspond to the 65536 code positions of the ISO 10646 Basic Multilingual Plane (BMP).

- **Note**: First 256 values of Wide_Character have the same name as defined for Character
Ada.Characters.Handling

• Character Classification
• Conversion Functions (both character and string)

Characters
  - Control
  - Graphic
  - Alphanumeric
    - Letters
    - Digits
  - Special Graphic

Character Handling

• function To_Lower (Item : in Character) return Character;
• function To_Upper (Item : in Character) return Character;
• function Is_Character (Item : in Wide_Character) return Boolean;

subtype ISO_646 is Character range Character'Val(0) .. Character'Val(127);

ASCII

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>NUL</td>
<td>SOH</td>
<td>STX</td>
<td>ETX</td>
<td>EOT</td>
<td>ENQ</td>
<td>ACK</td>
<td>BEL</td>
<td>BS</td>
<td>HT</td>
<td>LF</td>
<td>VT</td>
<td>FF</td>
<td>CR</td>
<td>SO</td>
<td>SI</td>
</tr>
<tr>
<td>DLE</td>
<td>DC1</td>
<td>DC2</td>
<td>DC3</td>
<td>DC4</td>
<td>NAK</td>
<td>SYN</td>
<td>ETB</td>
<td>CAN</td>
<td>EM</td>
<td>SUB</td>
<td>ESC</td>
<td>FS</td>
<td>GS</td>
<td>RS</td>
<td>US</td>
</tr>
<tr>
<td>SP</td>
<td>!</td>
<td>&quot;</td>
<td>#</td>
<td>$</td>
<td>%</td>
<td>&amp;</td>
<td>'</td>
<td>(</td>
<td>)</td>
<td>*</td>
<td>+</td>
<td>,</td>
<td>-</td>
<td>.</td>
<td>/</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td>E</td>
<td>F</td>
</tr>
<tr>
<td>@</td>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td>E</td>
<td>F</td>
<td>G</td>
<td>H</td>
<td>I</td>
<td>J</td>
<td>K</td>
<td>L</td>
<td>M</td>
<td>N</td>
<td>O</td>
</tr>
<tr>
<td>P</td>
<td>Q</td>
<td>R</td>
<td>S</td>
<td>T</td>
<td>U</td>
<td>V</td>
<td>W</td>
<td>X</td>
<td>Y</td>
<td>Z</td>
<td>[</td>
<td>\</td>
<td>]</td>
<td>^</td>
<td>_</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>;</td>
<td>&lt;</td>
<td>=</td>
<td>&gt;</td>
<td>?</td>
<td>@</td>
<td>#</td>
</tr>
<tr>
<td>a</td>
<td>b</td>
<td>c</td>
<td>d</td>
<td>e</td>
<td>f</td>
<td>g</td>
<td>h</td>
<td>i</td>
<td>j</td>
<td>k</td>
<td>l</td>
<td>m</td>
<td>n</td>
<td>o</td>
<td>p</td>
</tr>
<tr>
<td>q</td>
<td>r</td>
<td>s</td>
<td>t</td>
<td>u</td>
<td>v</td>
<td>w</td>
<td>x</td>
<td>y</td>
<td>z</td>
<td></td>
<td></td>
<td>(</td>
<td>)</td>
<td>~</td>
<td>DEL</td>
</tr>
</tbody>
</table>

Character Handling

• The predefined operators for the type Character are the same as for any enumeration type

• function Is_ISO_646 (Item : in Character) return Boolean;

• function To_ISO_646 (Item : in Character; Substitute : in ISO_646 := ' ') return ISO_646;
**Strings**

- A string is an array of characters (**static**)
- So, a string \( S \) consists of the characters: \( S[1], ..., S[n-1], S[n] \)

  \[
  S_1 \quad S_2 \quad \ldots \quad S_{n-1} \quad S_n
  \]

- A contiguous subset of the characters of \( S \) is called a **substring** of \( S \)
  - I.e., if \( 1 \leq i \leq j \leq n \) then \( S[i], S[i+1], ..., S[j] \) is a substring of \( S \)
- The **null string** contains no characters (""")

**Ada.Strings**

- **subtype** Positive is Integer range 1 .. Integer'Last;
- **type** String is array (Positive range <>) of Character;
- **type** Wide_String is array (Positive range <>) of Wide_Character;

**Basic Operations On Strings**

- **append**: adds a character to the end of a string
- **insert**: inserts a string in the middle of another string
- **delete**: deletes part of a string
- **concatenate**: joins two strings together
- **substring**: returns part of a string
- **find**: returns the position at which one string occurs within another, or whether it exists
- **length**: returns the number of characters in a string
- **equals**: tests two strings for equality

**Pre-defined Operations**

- Strings have the same operators as one-dimensional arrays i.e.
  - Concatenation operator &
  - Ordering operators <, <=, >, and >=

Question: **constant** String := "How many characters?";
-- Question'First = 1, Question'Last = 20
-- Question'Length = 20 (the number of characters)
Ask_Twice : String := Question & Question;
-- constrained to (1..40)
Ada.Strings.Fixed

• **procedure** Move (Source : **in** String; Target : **out** String; Drop : **in**
  Truncation := Error; Justify : **in**
  Alignment := Left; Pad : **in** Character := Space);

• **function** Insert (Source : **in** String;
  Before : **in** Positive; New_Item : **in**
  String) return String;

• **function** Delete (Source : **in** String;
  From : **in** Positive; Through : **in**
  Natural) return String;