Defining Floating Point Types

Floating-point numbers are used to represent real numbers, such as 1.2345678901234567890.

The three most common floating-point types are REAL4, REAL8, and REAL10.

REAL4 is a 4-byte single-precision floating-point variable.

REAL8 is an 8-byte double-precision floating-point variable.

REAL10 is a 10-byte extended-precision floating-point variable.

```
rVal1    REAL4    -1.2
rVal2    REAL8    3.2E-260
rVal3    REAL10   4.6E+4096

ShortArray  REAL4    20  DUP(0.0)
```

rVal1 REAL4 -1.2:

This line declares a single-precision floating-point variable named rVal1 and initializes it with the value -1.2. It uses 4 bytes of memory to store this value. Single-precision provides a good balance between precision and memory usage for most general-purpose calculations.

rVal2 REAL8 3.2E-260:

Here, we declare a double-precision floating-point variable named rVal2 and initialize it with the value 3.2E-260. This type uses 8 bytes of memory, providing higher precision for very small or very large numbers, as in scientific calculations.

rVal3 REAL10 4.6E+4096:

This line declares an extended-precision floating-point variable named rVal3 and initializes it with the value 4.6E+4096. REAL10 uses 10 bytes of memory, making it suitable for extremely high-precision calculations, especially in advanced scientific and mathematical applications where precision is critical.

ShortArray REAL4 20 DUP(0.0):

This declares an array named ShortArray of 20 single-precision floating-point numbers. Each element of the array is initialized to 0.0. The DUP (duplicate) operator is used to specify that you want 20 copies of the value 0.0 in the array.

These declarations allow you to work with floating-point numbers of different precisions and create arrays of such numbers, with varying levels of precision and memory usage depending on your specific needs in your assembly program.

Table 3-4 Standard Real Number Types.

Data Type	Significant Digits	Approximate Range
Short real	6	1.18×10^{-38} to 3.40×10^{38}
Long real	15	2.23×10^{-308} to 1.79×10^{308}
Extended-precision real	19	3.37×10^{-4932} to 1.18×10^{4932}

The DD, DQ, and DT directives can define also real numbers:

```
rVal1 DD -1.2                ;short real
rVal2 DQ 3.2E-260            ;long real
rVal3 DT 4.6E+4096           ;extended-precision real
```

****

The precision of a floating-point number refers to the number of significant digits that it can represent. Single-precision floating-point numbers have a precision of 7 significant digits. Double-precision floating-point numbers have a precision of 15 significant digits. Extended-precision floating-point numbers have a precision of 19 significant digits.

The range of a floating-point number refers to the set of values that it can represent. Single-precision floating-point numbers can represent values from approximately ±3.4E38 to ±1.2E-38. Double-precision floating-point numbers can represent values from approximately ±1.7E308 to ±2.4E-308. Extended-precision floating-point numbers can represent values from approximately ±4.9E324 to ±1.1E-324.

Here are some examples of how to declare and initialize floating-point variables in assembly language:

```
; Declare a single-precision floating-point variable.
rVal1 REAL4

; Initialize the variable to the value -1.2.
rVal1 = -1.2

; Declare a double-precision floating-point variable.
rVal2 REAL8

; Initialize the variable to the value 3.2E-260.
rVal2 = 3.2E-260

; Declare an extended-precision floating-point variable.
rVal3 REAL10

; Initialize the variable to the value 4.6E+4096.
rVal3 = 4.6E+4096

; Declare an array of 20 single-precision floating-point variables.
ShortArray REAL4 20 DUP(0.0)
```

Certainly, let's clarify the difference between "real numbers" and "floating-point numbers" in the context of MASM assembly:

- **Real Numbers:** In mathematics, a real number is a concept that represents a number with unlimited precision and size. Real numbers can include integers, fractions, irrational numbers (like pi), and many more. They are idealized mathematical entities without any practical limitations in terms of precision or range. In essence, real numbers are perfect mathematical abstractions.

- **Floating-Point Numbers:** In contrast, floating-point numbers used in computer programming, such as real4 and real8 in MASM, are not real numbers in the mathematical sense. Instead, they are approximations of real numbers that have limited precision and range due to the finite storage capacity of computers. These floating-point types can represent a wide range of numbers, including very large or very small values, but they do so with a fixed number of significant digits (precision). The accuracy of these representations is subject to the limitations of the hardware and the specific data type used.

So, when MASM uses data types like real4 and real8, it's important to understand that they are approximations of real numbers, and their precision and range are determined by the number of bits allocated for them. While they can represent a broad range of values, they may not capture the unlimited precision and size of true mathematical real numbers. Therefore, they are referred to as "floating-point" numbers in the context of computer programming to emphasize their approximate nature.