

# The Index Register

The Index register (Ri) is one of the most powerful programming tools available in the calculator. In addition to simple storage and recall of data, it can also be used for:

- Program loop counter and control functions (**[DSE]** and **[ISG]**).
- Indirectly addressing data storage registers, branches and subroutines.

## The Index Register For Program Looping

The value in the Index Register is interpreted with the format of a control number “±nnnnn.xxxyy”; “±nnnnn” is the current counter value, “xxx” is the counter test value and “yy” is the increment or decrement value.

“nnnnn” is the integer part of the control number in the Index Register. It is used for counting successive iterations of a program loop. The “nnnnn” value can be from 1 to 5 digits with the default value of zero.

“xxx” is in the decimal portion of the control number and is used by the **[DSE]** and **[ISG]** functions to increase or decrease the “nnnnn” counter value and test the result with “xxx”. The “xxx” value must be specified as a three-digits number.

“yy” is also in the decimal portion of the control number and tells to the **[DSE]** and **[ISG]** functions what increment use to increase or decrease the “nnnnn” counter value. The “yy” is a two digits number and if it is not specified the default value is 01.

**The [DSE] Function:** (Decrement and skip if equal or less than)

When a program is running and reach the DSE function the following actions are performed:

- Gets the “nnnnn”, “xxx” and “yy” portions from the index register.
- The value of “nnnnn” is decremented by “yy” and the result is updated in the index register (nnnnn.xxxyy).
- If “nnnnn” is equal or less than “xxx” then, the next program line is skipped.
- If “nnnnn” is greater than “xxx”, the next program line is executed.

## The **[ISG]** Function: (Increment and skip if greater than)

When a program is running and reach the ISG function the following actions are performed:

- Gets the “nnnnn”, “xxx” and “yy” portions from the index register.
- The value of “nnnnn” is incremented by “yy” and the result is updated in the index register (nnnnn.xxxyy).
- If “nnnnn” is greater than “xxx” then, the next program line is skipped.
- If “nnnnn” is less or equal than “xxx”, the next program line is executed.

## The Index Register For Indirect Addressing and Jumping

The integer part, “n”, of the value in the Index Register is used as a control number for the following functions:

“n”	<b>[GTO] [ I ]</b> or <b>[GSB] [ I ]</b>	<b>[STO] [(i)]</b> or <b>[RCL] [(i)]</b>
0 to 9	Transfer execution to <b>[LBL] 0 to 9</b>	Address storage register 0 to 9
10	Transfer execution to <b>[LBL] [A]</b>	Address storage register 10
11	Transfer execution to <b>[LBL] [B]</b>	Address storage register 11
12	Transfer execution to <b>[LBL] [C]</b>	Address storage register 12
13	Transfer execution to <b>[LBL] [D]</b>	Address storage register 13
14	Transfer execution to <b>[LBL] [E]</b>	Address storage register 14
15	—	Address storage register 15
16	—	Address storage register 16
17	—	Address storage register 17
18	—	Address storage register 18
19	—	Address storage register 19
20	—	Address index register itself
n < 0	in <b>[GTO]</b> jumps to program line “n”	Address register ABS(“n”)