**Author(s) :** F. Antonelli **Library:** MATHLIB
**Submitter :** F. Carminati **Submitted:** 29.05.1989
**Language :** Fortran (IBM: Assembler) **Revised:**

# Search Operations on Sparse Vectors

Performs logical search and data movement operations on sparse vectors. On Cray systems these routines are part of the default libraries (scilib). An optimized Assembler version is provided for IBM 3090 with Vector Facilities. Fortran code is used on the other systems.

**Structure:**

`SUBROUTINE` and `FUNCTION` subprograms
User Entry Names: `IILZ`, `ILSUM`, `SCATTER`, `GATHER`, `WHENEQ`, `WHENNE`, `WHENFLT`, `WHENFGT`, `WHENFLE`, `WHENFGE`, `WHENILT`, `WHENIGT`, `WHENILE`, `WHENIGE`

**Usage:**

The arguments in the calling sequences below are defined as follows:

| | |
|---|---|
| `A,B` | (`REAL`) One-dimensional arrays. |
| `IA,INDX` | (`INTEGER`) One-dimensional arrays. |
| `LA` | (`LOGICAL`) One-dimensional array. |
| `NW,INC` | (`INTEGER`) Variables or expressions. |
| `TARG` | (`REAL`) Variable or expression. |
| `ITARG,NFOUND` | (`INTEGER`) Variables. |

In any arithmetic expression,

```
IILZ(NW,A,INC)
```

represents the `INTEGER` number of leading zero elements in
$\mathtt{LA(1)}, \mathtt{LA(INC}+1), \mathtt{LA(2*INC}+1), \ldots, \mathtt{LA((NW}-1)*\mathtt{INC}+1)$;

```
ILSUM(NW,LA,INC)
```

represents the `INTEGER` number of `.TRUE.` elements in
$\mathtt{LA(1)}, \mathtt{LA(INC}+1), \mathtt{LA(2*INC}+1), \ldots, \mathtt{LA((NW}-1)*\mathtt{INC}+1)$.

```
CALL SCATTER(NW,A,INDX,B)
CALL GATHER(NW,A,B,INDX)
```

set $\mathtt{A(INDX(I))} = \mathtt{B(I)}$, $(\mathtt{I} = 1, 2, \ldots, \mathtt{NW})$ and $\mathtt{A(I)} = \mathtt{B(INDX(I))}$, $(\mathtt{I} = 1, 2, \ldots, \mathtt{NW})$, respectively.

```
CALL WHENFxx(NW,A,INC,TARG,INDX,NFOUND)
```

searches $\mathtt{A(1)}, \mathtt{A(INC}+1), \mathtt{A(2*INC}+1), \ldots, \mathtt{A((NW}-1)*\mathtt{INC}+1)$ for elements which satisfy the relation `A(.).xx.TARG` where $\mathtt{xx} = \mathtt{LT}, \mathtt{LE}, \mathtt{GT}, \mathtt{GE}$. On exit, $\mathtt{INDX(1)}, \ldots, \mathtt{INDX(NFOUND)}$ will contain the indices of the `NFOUND` elements which satisfy the relation specified.

```
CALL WHENIxx(NW,IA,INC,ITARG,INDX,NFOUND)
```

performes the same task as `WHENFxx` but for `INTEGER` draw and target.

```
CALL WHENEQ(NW,a,INC,targ,INDX,NFOUND)
CALL WHENNE(NW,a,INC,targ,INDX,NFOUND)
```

performs the same task as `WHENFxx` or `WHENIxx`, but for `xx` = `EQ`, `NE`, and `REAL` draw `a` and `REAL` target `targ`, or `INTEGER` draw `a` and `INTEGER` target `targ`, respectively.

•