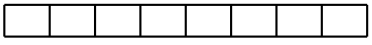
STAT 534 Handout 2

April 3, 2003

Static sets (arrays)

©Marina Meilă

mmp@stat.washington.edu

This is a "static" set: 

Operations on static sets:

- create
- delete
- iterate: iterates through values and performs the same operation on each

Creation (in C)

- static allocation (deletion implicit)
- dynamic allocation (pointers, deletion explicit)

Examples of statically allocated arrays:

```
double x[ 100 ];
double y[ 200 ][ 200 ];
integer years[ 10 ], number_students[ 10 ];
char thisname[ 25 ];
```

Some dynamically allocated arrays. They are not referred to by name but by a pointer.

```
int *pi;
double *pd, **ppd;
char *thisbuf;
int m = 12, n = 120, i, j;
double myfunction( int, int );
...
pi = (int *)calloc( n, sizeof( int ));
pd = (double *)malloc( m * sizeof( double ));
ppd = (double **) malloc( 100 * sizeof( double * ));
for( i = 0; i < 100; i++ ) {
  ppd[ i ] = (double *)calloc( m, sizeof( double ));
  for( j =0; j< m; j++ )
    ppd[ i ][ j ] = myfunction( i, j );
}
thisbuf = (char *)malloc( 1000 * sizeof( char ));
```

An iteration on an array (finding the maximum).

```
#define NMAX 1000 /* max size of x */

double max;
int imax;
double x[ NMAX ];
int n; /* actual size of x */
.....
for ( i = 0, max = 0; i < n; i++ )
  if( x[ i ] > max ) {
    max = x[ i ];
    imax = i;
}
```

Can this iteration be improved upon (corrected, in fact)? yes:

```
#define NMAX 1000 /* max size of x */

double max;
int imax;
double x[ NMAX ];
int n; /* actual size of x */
.....
for ( i = 1, max = x[ 0 ], imax = 0; i < n; i++ )
  if( x[ i ] > max ) {
    max = x[ i ];
    imax = i;
}
```

An example involving 2-dimensional arrays (Multiplication of square matrices)

```
double A[ 100 ][ 100 ], B[ 100 ][ 100 ], C[ 100 ][ 100 ];
int n; /* the dimension of A,B,C */
int i, j, k; /* indices */
.....
for( i = 0; i < n; i++ )
  for( j = 0; j < n; j ++ ) {
    C[ i ][ j ] = 0;
    for( k = 0; k < n; k ++ )
      C[ i ][ j ] += A[ i ][ k ]*B[ k ][ j ];
  }
```