

## [Help](#)

```
#include <stdio.h>
#include <stdlib.h>
#include <
href../../common/math/cdo/cdo_math_h_src.pdfmath.h>
#include "pnl/pnl_mathtools.h"

#define NRANSI
#define SWAP(a,b) temp=(a);(a)=(b);(b)=temp;
#define NSTACK 50

static int istack[NSTACK];

/* Sort function only used by the Marraquand Martineau algorithm for american
   options.

   bs1d, bs2d, bsnd
*/
void Sort(unsigned long n, double *arr)
{
    unsigned long i, ir = n, j, k, l = 1;
    int jstack = 0;
    unsigned int M = 7;
    double a, temp;

    for (;;)
    {
        if (ir - l < M)
        {
            for (j = l + 1; j <= ir; j++)
            {
                a = arr[j];
                for (i = j - 1; i >= l; i--)
                {
                    if (arr[i] <= a) break;
                    arr[i + 1] = arr[i];
                }
                arr[i + 1] = a;
            }
            if (jstack == 0) break;
        }
    }
}
```

```

        ir = istack[jstack--];
        l = istack[jstack--];
    }
else
{
    k = (l + ir) >> 1;
    SWAP(arr[k], arr[l + 1])
    if (arr[l + 1] > arr[ir])
    {
        SWAP(arr[l + 1], arr[ir])
    }
    if (arr[l] > arr[ir])
    {
        SWAP(arr[l], arr[ir])
    }
    if (arr[l + 1] > arr[l])
    {
        SWAP(arr[l + 1], arr[l])
    }
    i = l + 1;
    j = ir;
    a = arr[l];
    for (;;)
    {
        do i++;
        while (arr[i] < a);
        do j--;
        while (arr[j] > a);
        if (j < i) break;
        SWAP(arr[i], arr[j]);
    }
    arr[l] = arr[j];
    arr[j] = a;
    jstack += 2;
    if (jstack > NSTACK)
    {
        printf("SORTING ERROR\ n");
        exit(0);
    };
    if (ir - i + 1 >= j - 1)
    {

```

```

        istack[jstack] = ir;
        istack[jstack - 1] = i;
        ir = j - 1;
    }
else
{
    istack[jstack] = j - 1;
    istack[jstack - 1] = 1;
    l = i;
}
}

}
#undef NSTACK
#undef SWAP
#undef NRANSI

```