```
runner% cat testsort.c
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#define M 20
void exec_sort( void (* )(int *), int *);
void bubble_sort(int *);
void select_sort(int *);
int maxind(int , int *);
void swap(int *, int *);
void printit(int *);
void inita(int *);

void main(void)
{
   int a[M];
   void (* sortit[]) (int *) =
      {bubble_sort, select_sort};
   int i;
   for (i = 0; i < 2; i++) {
      printf("Sort # %d\n", i);
      exec_sort(*sortit[i], a);
   }
   printf("That's all folks\n");
}

void exec_sort( void (* sort)(int *), int *a)
{
   inita(a);
   sort(a);
   printit(a);
}

void bubble_sort(int *a)
{
   int dummy, i;
   for (dummy = 0; dummy < M-1; dummy++)
      for (i = 0; i < M-1; i++)
         if (a[i] > a[i+1])
            swap(&a[i], &a[i+1]);
}

void select_sort(int *a)
{
   int maxi, n = M-1;
   while (n > 0) {
      maxi = maxind(n, a);
      swap(&a[maxi], &a[n]);
      n--;
   }
}

int maxind(int m, int *a)
{
   int maxi;
   if (m == 0) return m;
   maxi = maxind(m-1, a);
   if (a[m] > a[maxi]) return m;
   return maxi;
}
void swap(int *x, int *y)
{
   int z = *x;
   *x = *y;
   *y = z;
}

void printit(int *a)
{
   int i;
   for (i = 0; i < 10; i++)
      printf("  %2i", a[i]);
   printf("\n");
}

void inita(int *a)
{
   int i;
   for (i = 0; i < M; i++)
      a[i] = (int)(1000.0*drand48());
}
runner% lint -m -u testsort.c
function returns value which is always ignored
    printf
runner% cc -o testsort testsort.c
runner% testsort
Sort # 0
  15  58  159  159  163  269  318  353  383  390
Sort # 1
   2  75  212  265  293  298  306  404  462  532
That's all folks
```

```
runner% cat testsort2.c
#include <stdio.h>
#include <stdlib.h>
#define M 20

struct function_sort {
   void (* sort) (int *);
   char *name;
};

void exec_sorts(struct function_sort sorts[], int *);

void bubble_sort(int *);
void select_sort(int *);
int maxind(int , int *);
void swap(int *, int *);
void printit(int *);
void inita(int *);

void main(void)
{
   int a[M];
   struct function_sort sorts[] =
      {{bubble_sort, "bubble sort"},
       {select_sort, "select sort"}};
   exec_sorts(sorts, a);
   printf("That's all folks\n");
}

void exec_sorts(struct function_sort sorts[], int *a)

{
   int i;
   for (i = 0; i < 2; i++) {
      inita(a);
      sorts[i].sort(a);
      printf("Sort: %s\n", sorts[i].name);
      printit(a);
   }
}

void bubble_sort(int *a)
{
   int dummy, i;
   for (dummy = 0; dummy < M-1; dummy++)
      for (i = 0; i < M-1; i++)
         if (a[i] > a[i+1])
            swap(&a[i], &a[i+1]);
}
void select_sort(int *a)
{
   int maxi, n = M-1;
   while (n > 0) {
      maxi = maxind(n, a);
      swap(&a[maxi], &a[n]);
      n--;
   }
}
int maxind(int m, int *a)
{
   int maxi;
   if (m == 0) return m;
   maxi = maxind(m-1, a);
   if (a[m] > a[maxi]) return m;
   return maxi;
}
void swap(int *x, int *y)
{
   int z = *x;
   *x = *y;
   *y = z;
}
void printit(int *a)
{
   int i;
   for (i = 0; i < 10; i++)
      printf("  %2i", a[i]);
   printf("\n");
}
void inita(int *a)
{
   int i;
   for (i = 0; i < M; i++)
      a[i] = (int)(1000.0*drand48());
}
runner% cc -o testsort2 testsort2.c
runner% testsort2
Sort: bubble sort
  15  58  159  159  163  269  318  353  383  390
Sort: select sort
   2  75  212  265  293  298  306  404  462  532
That's all folks
```