```
/*  segname.c - program to test crt0dat.asm startup and termination
**
**  Note that it appears MSC 7.0 reverses the order of data_seg
**  declarations. This may not be maintained in future versions.
**  have LINK generate a map so you can examine the order of the
**  segments
**  --Reinhold J. Gerharz, CompuServe ID 70662,2262.              */
*/

#include    <conio.h>

/* declare functions */
static void __near near_start(void);
static void __near near_end(void);
static void __far far_start(void);
static void __far far_end(void);

/* the order of the next four lines must not be changed */
#pragma data_seg("XIE", "DATA")
#pragma data_seg("XI", "DATA")
static int dummy1 = (int)near_start;
#pragma data_seg("XIB", "DATA")

/* the order of the next four lines must not be changed */
#pragma data_seg("XCE", "DATA")
#pragma data_seg("XC", "DATA")
static int dummy2 = (int)near_end;
#pragma data_seg("XCB", "DATA")

/* the order of the next four lines must not be changed */
#pragma data_seg("XIFE", "DATA")
#pragma data_seg("XIF", "DATA")
static long dummy3 = (long)far_start;
#pragma data_seg("XIFB", "DATA")

/* the order of the next four lines must not be changed */
#pragma data_seg("XCFE", "DATA")
#pragma data_seg("XCF", "DATA")
static long dummy4 = (long)far_end;
#pragma data_seg("XCFB", "DATA")

/* do not forget the next line before any other code */
#pragma data_seg()                      /* restore default segment */

int main()                              /* called third */
{
  cprintf("main() called\r\n");
  return 0;
}

static void __near near_start(void)   /* called second */
{
  cprintf("near_start called\r\n");
}

static void __near near_end(void)     /* called fourth */
{
  cprintf("near_end called\r\n");
}

static void __far far_start(void)       /* called first */
{
  cprintf("far_start called\r\n");
}

static void __far far_end(void)         /* called fifth */
{
  cprintf("far_end called\r\n");
}
```