```
////////////////////////////////////////////////////////////////////////////
//      var.c program to test memory allocation in C
//         M. Mizuno (c) 1995, 2004, 2005
//          modified for Learning Tree course 223P
//
// to compile var.c on Pentium,
// 1. execute Visual Studio .NET 2003 Command Prompt
// 2. go to the directory which contains var.c
// 3. issue cl /Od /FAcs var.c
//        /Od: disable optimization
//        /FAcs: generate a listing file with source code and machine code
///////////////////////////////////////////////////////////////////////////
long test(unsigned int ui, int i, short s, unsigned short us,
        char c, unsigned char uc, long l, unsigned long ul,
        int x, short y);

char ret;
int x=100;
static int si;
static int sj = 23;

int main(int argc, char **argv, char **envp)
{
        unsigned char uc;
        static short y = 99;
        short s;
        char c;
        unsigned short us;
        static int i;
        unsigned int ui;
        long l;
        unsigned long ul;

        if (i < 0) {
                ui = us + s − c;
        }
        else {
                ul = si − sj * 2;
        }

        while (sj > 0) {
                uc = y − 3;
                sj++;
        }

        ret = test(ui, i, s, us, c, uc, l, ul, x,  y);

        return 0;
}

long test(unsigned int ui, int i, short s, unsigned short us,
        char c, unsigned char uc, long l, unsigned long ul,
        int x, short y)
{
        char c1;
        int i1;
        char c2;

        ui = 1;
        i = 2;
        s = 3;
        us = 4;
        c = 5;
        uc = 6;
        l = 7;
        ul = 8;
        x = 9;
        y = 10;
        c1 = 11;
        c2 = 12;
        i1 = 13;
        return ui * 2 + l;
}
```

```
; Listing generated by Microsoft (R) Optimizing Compiler Version 19.12.25834.0

include listing.inc

INCLUDELIB LIBCMT
INCLUDELIB OLDNAMES

PUBLIC  x
_DATA   SEGMENT
COMM    ret:BYTE
_DATA   ENDS
_DATA   SEGMENT
x       DD      064H
sj      DD      017H
?y@?1??main@@9@9 DW 063H                                ; ‘main’::‘2’::y
_DATA   ENDS
PUBLIC  test
PUBLIC  main
_BSS    SEGMENT
si      DD      01H DUP (?)
?i@?1??main@@9@9 DD 01H DUP (?)                         ; ‘main’::‘2’::i
_BSS    ENDS
pdata   SEGMENT
$pdata$test DD  imagerel $LN3
        DD      imagerel $LN3+137
        DD      imagerel $unwind$test
$pdata$main DD  imagerel $LN7
        DD      imagerel $LN7+209
        DD      imagerel $unwind$main
pdata   ENDS
xdata   SEGMENT
$unwind$test DD 011801H
        DD      02218H
$unwind$main DD 011201H
        DD      0e212H
xdata   ENDS
; Function compile flags: /Odtp
; File c:\k-state\cis450\programs\memorytest\vc_x86_64\var.c
_TEXT   SEGMENT
uc$ = 80
c$ = 81
us$ = 84
s$ = 88
ul$ = 92
ui$ = 96
l$ = 100
argc$ = 128
argv$ = 136
envp$ = 144
main    PROC

; 23   : {

$LN7:
  00000 4c 89 44 24 18   mov     QWORD PTR [rsp+24], r8
  00005 48 89 54 24 10   mov     QWORD PTR [rsp+16], rdx
  0000a 89 4c 24 08      mov     DWORD PTR [rsp+8], ecx
  0000e 48 83 ec 78      sub     rsp, 120               ; 00000078H

; 24   :        unsigned char uc;
; 25   :        static short y = 99;
; 26   :        short s;
; 27   :        char c;
; 28   :        unsigned short us;
; 29   :        static int i;
; 30   :        unsigned int ui;
; 31   :        long l;
; 32   :        unsigned long ul;
; 33   :
; 34   :        if (i < 0) {

  00012 83 3d 00 00 00
        00 00            cmp     DWORD PTR ?i@?1??main@@9@9, 0
```

```
  00019 7d 19            jge     SHORT $LN4@main

; 35   :                ui = us + s - c;

  0001b 0f b7 44 24 54   movzx   eax, WORD PTR us$[rsp]
  00020 0f bf 4c 24 58   movsx   ecx, WORD PTR s$[rsp]
  00025 03 c1            add     eax, ecx
  00027 0f be 4c 24 51   movsx   ecx, BYTE PTR c$[rsp]
  0002c 2b c1            sub     eax, ecx
  0002e 89 44 24 60      mov     DWORD PTR ui$[rsp], eax

; 36   :        }

  00032 eb 16            jmp     SHORT $LN5@main
$LN4@main:

; 37   :        else {
; 38   :                ul = si - sj * 2;

  00034 8b 05 00 00 00
        00               mov     eax, DWORD PTR sj
  0003a d1 e0            shl     eax, 1
  0003c 8b 0d 00 00 00
        00               mov     ecx, DWORD PTR si
  00042 2b c8            sub     ecx, eax
  00044 8b c1            mov     eax, ecx
  00046 89 44 24 5c      mov     DWORD PTR ul$[rsp], eax
$LN5@main:
$LN2@main:

; 39   :        }
; 40   :
; 41   :        while (sj > 0) {

  0004a 83 3d 00 00 00
        00 00            cmp     DWORD PTR sj, 0
  00051 7e 1e            jle     SHORT $LN3@main

; 42   :                uc = y - 3;

  00053 0f bf 05 00 00
        00 00            movsx   eax, WORD PTR ?y@?1??main@@9@9
  0005a 83 e8 03         sub     eax, 3
  0005d 88 44 24 50      mov     BYTE PTR uc$[rsp], al

; 43   :                sj++;

  00061 8b 05 00 00 00
        00               mov     eax, DWORD PTR sj
  00067 ff c0            inc     eax
  00069 89 05 00 00 00
        00               mov     DWORD PTR sj, eax

; 44   :        }

  0006f eb d9            jmp     SHORT $LN2@main
$LN3@main:

; 45   :
; 46   :        ret = test(ui, i, s, us, c, uc, l, ul, x,  y);

  00071 0f b7 05 00 00
        00 00            movzx   eax, WORD PTR ?y@?1??main@@9@9
  00078 66 89 44 24 48   mov     WORD PTR [rsp+72], ax
  0007d 8b 05 00 00 00
        00               mov     eax, DWORD PTR x
  00083 89 44 24 40      mov     DWORD PTR [rsp+64], eax
  00087 8b 44 24 5c      mov     eax, DWORD PTR ul$[rsp]
  0008b 89 44 24 38      mov     DWORD PTR [rsp+56], eax
  0008f 8b 44 24 64      mov     eax, DWORD PTR l$[rsp]
  00093 89 44 24 30      mov     DWORD PTR [rsp+48], eax
  00097 0f b6 44 24 50   movzx   eax, BYTE PTR uc$[rsp]
  0009c 88 44 24 28      mov     BYTE PTR [rsp+40], al
  000a0 0f b6 44 24 51   movzx   eax, BYTE PTR c$[rsp]
  000a5 88 44 24 20      mov     BYTE PTR [rsp+32], al
  000a9 44 0f b7 4c 24
        54               movzx   r9d, WORD PTR us$[rsp]
  000af 44 0f b7 44 24
        58               movzx   r8d, WORD PTR s$[rsp]
  000b5 8b 15 00 00 00
        00               mov     edx, DWORD PTR ?i@?1??main@@9@9
  000bb 8b 4c 24 60      mov     ecx, DWORD PTR ui$[rsp]
  000bf e8 00 00 00 00   call    test
  000c4 88 05 00 00 00
        00               mov     BYTE PTR ret, al

; 47   :
; 48   :        return 0;

  000ca 33 c0            xor     eax, eax

; 49   : }

  000cc 48 83 c4 78      add     rsp, 120               ; 00000078H
  000d0 c3               ret     0
main    ENDP
_TEXT   ENDS
; Function compile flags: /Odtp
; File c:\k-state\cis450\programs\memorytest\vc_x86_64\var.c
_TEXT   SEGMENT
c1$ = 0
c2$ = 1
i1$ = 4
ui$ = 32
i$ = 40
s$ = 48
us$ = 56
c$ = 64
uc$ = 72
l$ = 80
ul$ = 88
x$ = 96
y$ = 104
test    PROC

; 54   : {

$LN3:
  00000 66 44 89 4c 24
        20               mov     WORD PTR [rsp+32], r9w
  00006 66 44 89 44 24
        18               mov     WORD PTR [rsp+24], r8w
  0000c 89 54 24 10      mov     DWORD PTR [rsp+16], edx
  00010 89 4c 24 08      mov     DWORD PTR [rsp+8], ecx
  00014 48 83 ec 18      sub     rsp, 24

; 55   :        char c1;
; 56   :        int i1;
; 57   :        char c2;
; 58   :
; 59   :        ui = 1;

  00018 c7 44 24 20 01
        00 00 00         mov     DWORD PTR ui$[rsp], 1

; 60   :        i = 2;

  00020 c7 44 24 28 02
        00 00 00         mov     DWORD PTR i$[rsp], 2

; 61   :        s = 3;

  00028 b8 03 00 00 00   mov     eax, 3
  0002d 66 89 44 24 30   mov     WORD PTR s$[rsp], ax

; 62   :        us = 4;
```

```

  00032 b8 04 00 00 00   mov     eax, 4
  00037 66 89 44 24 38   mov     WORD PTR us$[rsp], ax

; 63   :        c = 5;

  0003c c6 44 24 40 05   mov     BYTE PTR c$[rsp], 5

; 64   :        uc = 6;

  00041 c6 44 24 48 06   mov     BYTE PTR uc$[rsp], 6

; 65   :        l = 7;

  00046 c7 44 24 50 07
        00 00 00         mov     DWORD PTR l$[rsp], 7

; 66   :        ul = 8;

  0004e c7 44 24 58 08
        00 00 00         mov     DWORD PTR ul$[rsp], 8

; 67   :        x = 9;

  00056 c7 44 24 60 09
        00 00 00         mov     DWORD PTR x$[rsp], 9

; 68   :        y = 10;

  0005e b8 0a 00 00 00   mov     eax, 10
  00063 66 89 44 24 68   mov     WORD PTR y$[rsp], ax

; 69   :        c1 = 11;

  00068 c6 04 24 0b      mov     BYTE PTR c1$[rsp], 11

; 70   :        c2 = 12;

  0006c c6 44 24 01 0c   mov     BYTE PTR c2$[rsp], 12

; 71   :        i1 = 13;

  00071 c7 44 24 04 0d
        00 00 00         mov     DWORD PTR i1$[rsp], 13

; 72   :        return ui * 2 + l;

  00079 8b 44 24 50      mov     eax, DWORD PTR l$[rsp]
  0007d 8b 4c 24 20      mov     ecx, DWORD PTR ui$[rsp]
  00081 8d 04 48         lea     eax, DWORD PTR [rax+rcx*2]

; 73   : }

  00084 48 83 c4 18      add     rsp, 24
  00088 c3               ret     0
test    ENDP
_TEXT   ENDS
END
```