

; Listing generated by Microsoft (R) Optimizing Compiler Version 15.00.30729.01

```
TITLE C:\work\ml\regression3.c
.686P
.XMM
include listing.inc
.model flat
```

```
INCLUDELIB LIBCMT
INCLUDELIB OLDNAMES
```

```
PUBLIC _x_values
PUBLIC _y_values
PUBLIC _numval
PUBLIC _log_x
PUBLIC _log_y
PUBLIC _regress_name
PUBLIC _a1_weight
PUBLIC _b1_weight
PUBLIC _print_row
```

```
_BSS SEGMENT
$SG5122 DB 01H DUP (?)
```

```
_BSS ENDS
_DATA SEGMENT
COMM _sumx2:QWORD
COMM _a1:QWORD
COMM _sumy2:QWORD
COMM _b1:QWORD
COMM _sumxy:QWORD
COMM _sumx:QWORD
COMM _sumy:QWORD
COMM _r2:QWORD
```

```
_DATA ENDS
_BSS SEGMENT
ALIGN 8
```

```
_x_values DQ 02800H DUP (?)
_y_values DQ 02804H DUP (?)
_numval DD 01H DUP (?)
ALIGN 8
```

```
_log_x DQ 02800H DUP (?)
_log_y DQ 02804H DUP (?)
_print_row DD 01H DUP (?)
```

```
_BSS ENDS
_DATA SEGMENT
$SG5057 DB 'x', 09H, 09H, 'log(x)', 09H, 09H, 'y', 09H, 09H, 'log(y)'
DB 0aH, 00H
ORG $+2
$SG5061 DB '%f', 09H, '%f', 09H, '%f', 09H, '%f', 0aH, 00H
ORG $+3
$SG5062 DB 0aH, 00H
ORG $+2
$SG5063 DB 'sumx = %f', 0aH, 00H
ORG $+1
$SG5064 DB 'sumy = %f', 0aH, 00H
ORG $+1
$SG5065 DB 'sumx2 = %f', 0aH, 00H
$SG5066 DB 'sumy2 = %f', 0aH, 00H
$SG5067 DB 'sumxy = %f', 0aH, 00H
$SG5123 DB 'pow', 00H
$SG5124 DB 'log', 00H
$SG5125 DB 'exp', 00H
$SG5126 DB 'lin', 00H
_regress_name DD FLAT:$SG5122
```

```
DD FLAT:$SG5123
DD FLAT:$SG5124
DD FLAT:$SG5125
DD FLAT:$SG5126
$SG5159 DB ' ', 09H, 00H
ORG $+1
$SG5171 DB 'Too many data points', 00H
ORG $+3
$SG5175 DB 'Missing data', 00H
ORG $+3
_a1_weight DQ 03ff000000000000r ; 1
_b1_weight DQ 03ff000000000000r ; 1
$SG5211 DB '%f <= %f * pow(%f, %f)', 0aH, 00H
$SG5213 DB '%f', 09H, '%f', 09H, '%f', 09H, '%f', 09H, '%f', 0aH, 00H
$SG5214 DB '[%.04f, %.04f] [%.04f, %.04f] total error = %.04f', 0aH, 00H
ORG $+1
$SG5219 DB 'converged', 00H
ORG $+2
$SG5234 DB '[:s: a=%f b=%f r2=%.16f]', 0aH, 00H
ORG $+2
$SG5240 DB 'Regressions didn''t work', 0aH, 00H
ORG $+3
$SG5246 DB 'y = %f (many->one)', 0aH, 00H
$SG5252 DB 'x = %f (one->many)', 0aH, 00H
$SG5253 DB '(r2 = %f)', 0aH, 00H
ORG $+1
$SG5254 DB '%s: ', 00H
ORG $+3
$SG5260 DB 'y=%f+%fx', 0aH, 00H
ORG $+2
$SG5261 DB 'double f(double x) { return %f + (%f * x); }', 0aH, 00H
ORG $+2
$SG5263 DB 'y=%f+%fln(x)', 0aH, 00H
ORG $+2
$SG5264 DB 'double f(double x) { return %f + (%f * log(x)); }', 0aH, 00H
ORG $+1
$SG5265 DB 'double f(double x) { return %f + (%f * log10(x)); }', 0aH
DB 00H
ORG $+3
$SG5267 DB 'y=%fln^(%fx)', 0aH, 00H
ORG $+2
$SG5268 DB 'double f(double x) { return %f * exp(%f * x); }', 0aH, 00H
ORG $+3
$SG5270 DB 'y=%fx^%f', 0aH, 00H
ORG $+2
$SG5271 DB 'double f(double x) { return %f * pow(x, %f); }', 0aH, 00H
$SG5272 DB 0aH, 00H
_DATA ENDS
PUBLIC _stats
EXTRN __printf:PROC
EXTRN __fltused:DWORD
; Function compile flags: /Odtp
_TEXT SEGMENT
_i$ = -4 ; size = 4
_stats PROC
; File c:\work\ml\regression3.c
; Line 41
push ebp
mov ebp, esp
push ecx
; Line 43
push OFFSET $SG5057
call __printf
add esp, 4
; Line 44
```

```
    mov     DWORD PTR _i$[ebp], 0
    jmp     SHORT $LN3@stats
$LN2@stats:
    mov     eax, DWORD PTR _i$[ebp]
    add     eax, 1
    mov     DWORD PTR _i$[ebp], eax
$LN3@stats:
    mov     ecx, DWORD PTR _i$[ebp]
    cmp     ecx, DWORD PTR numval
    jge     SHORT $LN1@stats
; Line 47
    mov     edx, DWORD PTR _i$[ebp]
    sub     esp, 8
    fld     QWORD PTR _log_y[edx*8]
    fstp   QWORD PTR [esp]
    mov     eax, DWORD PTR _i$[ebp]
    sub     esp, 8
    fld     QWORD PTR _y_values[eax*8]
    fstp   QWORD PTR [esp]
    mov     ecx, DWORD PTR _i$[ebp]
    sub     esp, 8
    fld     QWORD PTR _log_x[ecx*8]
    fstp   QWORD PTR [esp]
    mov     edx, DWORD PTR _i$[ebp]
    sub     esp, 8
    fld     QWORD PTR _x_values[edx*8]
    fstp   QWORD PTR [esp]
    push   OFFSET $SG5061
    call   _printf
    add     esp, 36 ; 00000024H
    jmp     SHORT $LN2@stats
$LN1@stats:
; Line 48
    push   OFFSET $SG5062
    call   _printf
; Line 49
    add     esp, -4 ; ffffffffch
    fld     QWORD PTR _sumx
    fstp   QWORD PTR [esp]
    push   OFFSET $SG5063
    call   _printf
; Line 50
    add     esp, 4
    fld     QWORD PTR _sumy
    fstp   QWORD PTR [esp]
    push   OFFSET $SG5064
    call   _printf
; Line 51
    add     esp, 4
    fld     QWORD PTR _sumx2
    fstp   QWORD PTR [esp]
    push   OFFSET $SG5065
    call   _printf
; Line 52
    add     esp, 4
    fld     QWORD PTR _sumy2
    fstp   QWORD PTR [esp]
    push   OFFSET $SG5066
    call   printf
; Line 53
    add     esp, 4
    fld     QWORD PTR _sumxy
    fstp   QWORD PTR [esp]
    push   OFFSET $SG5067
    call   _printf
```

```
    add     esp, 12                                ; 0000000cH
; Line 54
    mov     esp, ebp
    pop     ebp
    ret     0
_stats    ENDP
_TEXT     ENDS
PUBLIC   __real@0000000000000000
PUBLIC   do_linear
;         COMDAT __real@0000000000000000
CONST    SEGMENT
__real@0000000000000000 DQ 0000000000000000r    ; 0
; Function compile flags: /Odtp
CONST    ENDS
TEXT     SEGMENT
_w$ = -4                                       ; size = 4
_x$ = 8                                       ; size = 4
_y$ = 12                                      ; size = 4
_n$ = 16                                      ; size = 4
_do_linear PROC
; Line 57
    push   ebp
    mov    ebp, esp
    push   ecx
    push   esi
; Line 59
    fldz
    fstp   QWORD PTR _sumxy
    fld   QWORD PTR _sumxy
    fstp   QWORD PTR _sumy2
    fld   QWORD PTR _sumy2
    fstp   QWORD PTR _sumy
    fld   QWORD PTR _sumy
    fstp   QWORD PTR _sumx2
    fld   QWORD PTR _sumx2
    fstp   QWORD PTR _sumx
; Line 60
    mov    DWORD PTR _w$[ebp], 0
    jmp   SHORT $LN3@do_linear
$LN2@do_linear:
    mov    eax, DWORD PTR _w$[ebp]
    add   eax, 1
    mov    DWORD PTR _w$[ebp], eax
$LN3@do_linear:
    mov    ecx, DWORD PTR _w$[ebp]
    cmp   ecx, DWORD PTR _n$[ebp]
    jge   $LN1@do_linear
; Line 62
    mov    edx, DWORD PTR _w$[ebp]
    mov    eax, DWORD PTR _x$[ebp]
    fld   QWORD PTR _sumx
    fadd  QWORD PTR [eax+edx*8]
    fstp  QWORD PTR _sumx
; Line 63
    mov    ecx, DWORD PTR _w$[ebp]
    mov    edx, DWORD PTR _x$[ebp]
    mov    eax, DWORD PTR _w$[ebp]
    mov    esi, DWORD PTR _x$[ebp]
    fld   QWORD PTR [edx+ecx*8]
    fmul  QWORD PTR [esi+eax*8]
    fadd  QWORD PTR _sumx2
    fstp  QWORD PTR _sumx2
; Line 64
    mov    ecx, DWORD PTR _w$[ebp]
    mov    edx, DWORD PTR _y$[ebp]
```

```
fld QWORD PTR _sumy
fadd QWORD PTR [edx+ecx*8]
fstp QWORD PTR _sumy
; Line 65
mov eax, DWORD PTR _w$[ebp]
mov ecx, DWORD PTR _y$[ebp]
mov edx, DWORD PTR _w$[ebp]
mov esi, DWORD PTR _y$[ebp]
fld QWORD PTR [ecx+eax*8]
fmul QWORD PTR [esi+edx*8]
fadd QWORD PTR _sumy2
fstp QWORD PTR _sumy2
; Line 66
mov eax, DWORD PTR _w$[ebp]
mov ecx, DWORD PTR _x$[ebp]
mov edx, DWORD PTR _w$[ebp]
mov esi, DWORD PTR _y$[ebp]
fld QWORD PTR [ecx+eax*8]
fmul QWORD PTR [esi+edx*8]
fadd QWORD PTR _sumxy
fstp QWORD PTR _sumxy
; Line 67
jmp $LN2@do_linear
$LN1@do_linear:
; Line 68
pop esi
mov esp, ebp
pop ebp
ret 0
_do_linear ENDP
_TEXT ENDS
PUBLIC _do_log
; Function compile flags: /Odtp
TEXT SEGMENT
_lx$ = -16 ; size = 8
_w$ = -4 ; size = 4
_x$ = 8 ; size = 4
_y$ = 12 ; size = 4
_n$ = 16 ; size = 4
_do_log PROC
; Line 72
push ebp
mov ebp, esp
sub esp, 16 ; 00000010H
push esi
; Line 75
fldz
fstp QWORD PTR _sumxy
fld QWORD PTR _sumxy
fstp QWORD PTR _sumy2
fld QWORD PTR _sumy2
fstp QWORD PTR _sumy
fld QWORD PTR _sumy
fstp QWORD PTR _sumx2
fld QWORD PTR _sumx2
fstp QWORD PTR _sumx
; Line 76
mov DWORD PTR _w$[ebp], 0
jmp SHORT $LN3@do_log
$LN2@do_log:
mov eax, DWORD PTR _w$[ebp]
add eax, 1
mov DWORD PTR _w$[ebp], eax
$LN3@do_log:
mov ecx, DWORD PTR _w$[ebp]
```

```
    cmp     ecx, DWORD PTR _n$[ebp]
    jge     SHORT $LN1@do_log
; Line 78
    mov     edx, DWORD PTR _w$[ebp]
    fld     QWORD PTR _log_x[edx*8]
    fstp   QWORD PTR _lx$[ebp]
; Line 79
    fld     QWORD PTR _sumx
    fadd   QWORD PTR _lx$[ebp]
    fstp   QWORD PTR _sumx
; Line 80
    fld     QWORD PTR _lx$[ebp]
    fmul   QWORD PTR _lx$[ebp]
    fadd   QWORD PTR _sumx2
    fstp   QWORD PTR _sumx2
; Line 81
    mov     eax, DWORD PTR _w$[ebp]
    mov     ecx, DWORD PTR _y$[ebp]
    fld     QWORD PTR _sumy
    fadd   QWORD PTR [ecx+eax*8]
    fstp   QWORD PTR _sumy
; Line 82
    mov     edx, DWORD PTR _w$[ebp]
    mov     eax, DWORD PTR _y$[ebp]
    mov     ecx, DWORD PTR _w$[ebp]
    mov     esi, DWORD PTR _y$[ebp]
    fld     QWORD PTR [eax+edx*8]
    fmul   QWORD PTR [esi+ecx*8]
    fadd   QWORD PTR _sumy2
    fstp   QWORD PTR _sumy2
; Line 83
    mov     edx, DWORD PTR _w$[ebp]
    mov     eax, DWORD PTR _y$[ebp]
    fld     QWORD PTR _lx$[ebp]
    fmul   QWORD PTR [eax+edx*8]
    fadd   QWORD PTR _sumxy
    fstp   QWORD PTR _sumxy
; Line 84
    jmp     $LN2@do_log
$LN1@do_log:
; Line 85
    pop     esi
    mov     esp, ebp
    pop     ebp
    ret     0
_do_log ENDP
_TEXT ENDS
PUBLIC _do_exp
; Function compile flags: /Odtp
_TEXT SEGMENT
_ly$ = -16 ; size = 8
_w$ = -4 ; size = 4
_x$ = 8 ; size = 4
_y$ = 12 ; size = 4
_n$ = 16 ; size = 4
_do_exp PROC
; Line 89
    push   ebp
    mov    ebp, esp
    sub    esp, 16 ; 00000010H
    push   esi
; Line 92
    fldz
    fstp   QWORD PTR _sumxy
    fld    QWORD PTR _sumxy
```

```
    fstp    QWORD PTR _sumy2
    fld     QWORD PTR _sumy2
    fstp    QWORD PTR _sumy
    fld     QWORD PTR _sumy
    fstp    QWORD PTR _sumx2
    fld     QWORD PTR _sumx2
    fstp    QWORD PTR _sumx
; Line 93
    mov     DWORD PTR w$[ebp], 0
    jmp     SHORT $LN3@do_exp
$LN2@do_exp:
    mov     eax, DWORD PTR _w$[ebp]
    add     eax, 1
    mov     DWORD PTR _w$[ebp], eax
$LN3@do_exp:
    mov     ecx, DWORD PTR _w$[ebp]
    cmp     ecx, DWORD PTR _n$[ebp]
    jge     SHORT $LN1@do_exp
; Line 95
    mov     edx, DWORD PTR _w$[ebp]
    fld     QWORD PTR _log_y[edx*8]
    fstp    QWORD PTR _ly$[ebp]
; Line 96
    mov     eax, DWORD PTR _w$[ebp]
    mov     ecx, DWORD PTR _x$[ebp]
    fld     QWORD PTR _sumx
    fadd    QWORD PTR [ecx+eax*8]
    fstp    QWORD PTR _sumx
; Line 97
    mov     edx, DWORD PTR _w$[ebp]
    mov     eax, DWORD PTR _x$[ebp]
    mov     ecx, DWORD PTR _w$[ebp]
    mov     esi, DWORD PTR _x$[ebp]
    fld     QWORD PTR [eax+edx*8]
    fmul   QWORD PTR [esi+ecx*8]
    fadd    QWORD PTR _sumx2
    fstp    QWORD PTR _sumx2
; Line 98
    fld     QWORD PTR _sumy
    fadd    QWORD PTR _ly$[ebp]
    fstp    QWORD PTR _sumy
; Line 99
    fld     QWORD PTR _ly$[ebp]
    fmul   QWORD PTR _ly$[ebp]
    fadd    QWORD PTR _sumy2
    fstp    QWORD PTR _sumy2
; Line 100
    mov     edx, DWORD PTR _w$[ebp]
    mov     eax, DWORD PTR _x$[ebp]
    fld     QWORD PTR [eax+edx*8]
    fmul   QWORD PTR _ly$[ebp]
    fadd    QWORD PTR _sumxy
    fstp    QWORD PTR _sumxy
; Line 101
    jmp     $LN2@do_exp
$LN1@do_exp:
; Line 102
    pop     esi
    mov     esp, ebp
    pop     ebp
    ret     0
_do_exp ENDP
_TEXT ENDS
PUBLIC _do_pow
; Function compile flags: /Odtg
```

```
_TEXT SEGMENT
_w$ = -4 ; size = 4
_x$ = 8 ; size = 4
_y$ = 12 ; size = 4
_n$ = 16 ; size = 4
_do_pow PROC
; Line 116
    push    ebp
    mov     ebp, esp
    push    ecx
; Line 118
    fldz
    fstp   QWORD PTR _sumxy
    fld   QWORD PTR _sumxy
    fstp   QWORD PTR _sumy2
    fld   QWORD PTR _sumy2
    fstp   QWORD PTR _sumy
    fld   QWORD PTR _sumy
    fstp   QWORD PTR _sumx2
    fld   QWORD PTR _sumx2
    fstp   QWORD PTR _sumx
; Line 119
    mov     DWORD PTR _w$[ebp], 0
    jmp     $LN3@do_pow
$LN2@do_pow:
    mov     eax, DWORD PTR _w$[ebp]
    add     eax, 1
    mov     DWORD PTR _w$[ebp], eax
$LN3@do_pow:
    mov     ecx, DWORD PTR _w$[ebp]
    cmp     ecx, DWORD PTR _n$[ebp]
    jge     $LN1@do_pow
; Line 121
    mov     edx, DWORD PTR w$[ebp]
    fld   QWORD PTR _sumx
    fadd   QWORD PTR _log_x[edx*8]
    fstp   QWORD PTR _sumx
; Line 122
    mov     eax, DWORD PTR _w$[ebp]
    mov     ecx, DWORD PTR w$[ebp]
    fld   QWORD PTR _log_x[eax*8]
    fmul   QWORD PTR _log_x[ecx*8]
    fadd   QWORD PTR _sumx2
    fstp   QWORD PTR _sumx2
; Line 123
    mov     edx, DWORD PTR _w$[ebp]
    fld   QWORD PTR _sumy
    fadd   QWORD PTR _log_y[edx*8]
    fstp   QWORD PTR _sumy
; Line 124
    mov     eax, DWORD PTR _w$[ebp]
    mov     ecx, DWORD PTR w$[ebp]
    fld   QWORD PTR _log_y[eax*8]
    fmul   QWORD PTR _log_y[ecx*8]
    fadd   QWORD PTR _sumy2
    fstp   QWORD PTR _sumy2
; Line 125
    mov     edx, DWORD PTR _w$[ebp]
    mov     eax, DWORD PTR w$[ebp]
    fld   QWORD PTR _log_x[edx*8]
    fmul   QWORD PTR _log_y[eax*8]
    fadd   QWORD PTR _sumxy
    fstp   QWORD PTR _sumxy
; Line 126
    jmp     $LN2@do_pow
```



```
$LN1@do_pow:
; Line 127
    mov     esp, ebp
    pop     ebp
    ret     0
_do_pow ENDP
_TEXT ENDS
PUBLIC _do_regress
; Function compile flags: /Odtp
_TEXT SEGMENT
tv64 = -4 ; size = 4
_type$ = 8 ; size = 4
_x$ = 12 ; size = 4
_y$ = 16 ; size = 4
_n$ = 20 ; size = 4
_do_regress PROC
; Line 147
    push    ebp
    mov     ebp, esp
    push    ecx
; Line 148
    mov     eax, DWORD PTR _type$[ebp]
    mov     DWORD PTR tv64[ebp], eax
    mov     ecx, DWORD PTR tv64[ebp]
    sub     ecx, 1
    mov     DWORD PTR tv64[ebp], ecx
    cmp     DWORD PTR tv64[ebp], 3
    ja     $LN5@do_regress
    mov     edx, DWORD PTR tv64[ebp]
    jmp     DWORD PTR $LN9@do_regress[edx*4]
$LN4@do_regress:
; Line 150
    mov     eax, DWORD PTR _n$[ebp]
    push    eax
    mov     ecx, DWORD PTR _y$[ebp]
    push    ecx
    mov     edx, DWORD PTR _x$[ebp]
    push    edx
    call    _do_linear
    add     esp, 12 ; 0000000cH
    jmp     SHORT $LN5@do_regress
$LN3@do_regress:
; Line 151
    mov     eax, DWORD PTR _n$[ebp]
    push    eax
    mov     ecx, DWORD PTR _y$[ebp]
    push    ecx
    mov     edx, DWORD PTR _x$[ebp]
    push    edx
    call    _do_log
    add     esp, 12 ; 0000000cH
    jmp     SHORT $LN5@do_regress
$LN2@do_regress:
; Line 152
    mov     eax, DWORD PTR _n$[ebp]
    push    eax
    mov     ecx, DWORD PTR _y$[ebp]
    push    ecx
    mov     edx, DWORD PTR x$[ebp]
    push    edx
    call    _do_exp
    add     esp, 12 ; 0000000cH
    jmp     SHORT $LN5@do_regress
$LN1@do_regress:
; Line 153
```

```
    mov     eax, DWORD PTR _n$[ebp]
    push   eax
    mov     ecx, DWORD PTR _y$[ebp]
    push   ecx
    mov     edx, DWORD PTR _x$[ebp]
    push   edx
    call   _do_pow
    add     esp, 12                                ; 0000000cH
$LN5@do_regress:
; Line 156
    fld     QWORD PTR _sumxy
    fld     QWORD PTR _sumx
    fld     QWORD PTR _sumy
    fsubp  ST(1), ST(0)
    fld     DWORD PTR _n$[ebp]
    fmul   QWORD PTR _sumx2
    fld     QWORD PTR _sumx
    fmul   QWORD PTR _sumx
    fsubp  ST(1), ST(0)
    fdivp  ST(1), ST(0)
    fstp   QWORD PTR _b1
; Line 157
    fld     QWORD PTR _b1
    fmul   QWORD PTR _sumx
    fsubr  QWORD PTR _sumy
    fidiv  DWORD PTR _n$[ebp]
    fstp   QWORD PTR _a1
; Line 158
    fld     DWORD PTR _n$[ebp]
    fmul   QWORD PTR _sumxy
    fld     QWORD PTR _sumx
    fmul   QWORD PTR _sumy
    fsubp  ST(1), ST(0)
    fmul   QWORD PTR _b1
    fld     DWORD PTR _n$[ebp]
    fmul   QWORD PTR _sumy2
    fld     QWORD PTR _sumy
    fmul   QWORD PTR _sumy
    fsubp  ST(1), ST(0)
    fdivp  ST(1), ST(0)
    fstp   QWORD PTR _r2
; Line 159
    mov     eax, DWORD PTR _type$[ebp]
; Line 160
    mov     esp, ebp
    pop    ebp
    ret    0
    npad   3
$LN9@do_regress:
    DD     $LN1@do_regress
    DD     $LN3@do_regress
    DD     $LN2@do_regress
    DD     $LN4@do_regress
_do_regress ENDP
_TEXT ENDS
PUBLIC _fail
EXTRN _exit:PROC
EXTRN puts:PROC
; Function compile flags: /Odtp
_TEXT SEGMENT
_s$ = 8                                ; size = 4
_fail PROC
; Line 162
    push   ebp
```

```
    mov     ebp, esp
    mov     eax, DWORD PTR _s$[ebp]
    push   eax
    call   _puts
    add    esp, 4
    push   1
    call   _exit
$LN2@fail:
    pop    ebp
    ret    0
_fail    ENDP
_TEXT   ENDS
PUBLIC  __$ArrayPad$
PUBLIC  _setxyvals
EXTRN  _log:PROC
EXTRN  _atof:PROC
EXTRN  _strtok:PROC
EXTRN  _isspace:PROC
EXTRN  _gets:PROC
EXTRN  ___security_cookie:DWORD
EXTRN  @__security_check_cookie@4:PROC
; Function compile flags: /Odt
_TEXT  SEGMENT
_linenum$ = -1068 ; size = 4
_buf$ = -1064 ; size = 1024
__$ArrayPad$ = -36 ; size = 4
_delim$ = -32 ; size = 4
_s$ = -28 ; size = 4
_x$ = -24 ; size = 8
_y$ = -16 ; size = 8
_xstr$ = -8 ; size = 4
_ystr$ = -4 ; size = 4
_setxyvals PROC
; Line 165
    push   ebp
    mov    ebp, esp
    sub    esp, 1068 ; 0000042cH
    mov    eax, DWORD PTR ___security_cookie
    xor    eax, ebp
    mov    DWORD PTR  $ArrayPad$[ebp], eax
; Line 168
    mov    DWORD PTR _delim$[ebp], OFFSET $SG5159
; Line 169
    mov    DWORD PTR _linenum$[ebp], 0
$LN15@setxyvals:
; Line 170
    lea   eax, DWORD PTR _buf$[ebp]
    push  eax
    call  _gets
    add   esp, 4
    mov   eax, DWORD PTR _s$[ebp], eax
    cmp   eax, DWORD PTR _s$[ebp], 0
    je    $LN9@setxyvals
; Line 172
    mov   ecx, DWORD PTR _linenum$[ebp]
    add   ecx, 1
    mov   DWORD PTR _linenum$[ebp], ecx
$LN8@setxyvals:
; Line 173
    mov   edx, DWORD PTR _s$[ebp]
    movsx eax, BYTE PTR [edx]
    push  eax
    call  _isspace
    add   esp, 4
    test  eax, eax
```

```
    je     SHORT $LN7@setxyvals
    mov    ecx, DWORD PTR _s$[ebp]
    add    ecx, 1
    mov    DWORD PTR _s$[ebp], ecx
    jmp    SHORT $LN8@setxyvals
$LN7@setxyvals:
; Line 174
    mov    edx, DWORD PTR _s$[ebp]
    movsx  eax, BYTE PTR [edx]
    test   eax, eax
    jne    SHORT $LN6@setxyvals
    jmp    SHORT $LN15@setxyvals
$LN6@setxyvals:
; Line 175
    mov    ecx, DWORD PTR  s$[ebp]
    movsx  edx, BYTE PTR [ecx]
    cmp    edx, 59 ; 0000003bH
    jne    SHORT $LN5@setxyvals
    jmp    SHORT $LN15@setxyvals
$LN5@setxyvals:
; Line 176
    mov    eax, DWORD PTR _s$[ebp]
    movsx  ecx, BYTE PTR [eax]
    cmp    ecx, 35 ; 00000023H
    jne    SHORT $LN4@setxyvals
    jmp    SHORT $LN15@setxyvals
$LN4@setxyvals:
; Line 177
    cmp    DWORD PTR _numval, 10240 ; 00002800H
    jl     SHORT $LN3@setxyvals
; Line 178
    push  OFFSET $SG5171
    call  _fail
    add   esp, 4
$LN3@setxyvals:
; Line 179
    mov    edx, DWORD PTR __delim$[ebp]
    push  edx
    mov    eax, DWORD PTR _s$[ebp]
    push  eax
    call  _strtok
    add   esp, 8
    mov    DWORD PTR _xstr$[ebp], eax
; Line 180
    mov    ecx, DWORD PTR __delim$[ebp]
    push  ecx
    push  0
    call  _strtok
    add   esp, 8
    mov    DWORD PTR _ystr$[ebp], eax
; Line 181
    cmp    DWORD PTR _xstr$[ebp], 0
    je     SHORT $LN1@setxyvals
    cmp    DWORD PTR _ystr$[ebp], 0
    je     SHORT $LN1@setxyvals
    mov    edx, DWORD PTR _ystr$[ebp]
    movsx  eax, BYTE PTR [edx]
    cmp    eax, 59 ; 0000003bH
    jne    SHORT $LN2@setxyvals
$LN1@setxyvals:
; Line 182
    push  OFFSET $SG5175
    call  _fail
    add   esp, 4
$LN2@setxyvals:
```

```
; Line 183
    mov     ecx, DWORD PTR _xstr$[ebp]
    push   ecx
    call   _atof
    add    esp, 4
    fstp   QWORD PTR _x$[ebp]
; Line 184
    mov     edx, DWORD PTR _ystr$[ebp]
    push   edx
    call   _atof
    add    esp, 4
    fstp   QWORD PTR _y$[ebp]
; Line 185
    mov     eax, DWORD PTR _numval
    fld    QWORD PTR x$[ebp]
    fstp   QWORD PTR _x_values[eax*8]
; Line 186
    mov     ecx, DWORD PTR _numval
    fld    QWORD PTR _y$[ebp]
    fstp   QWORD PTR _y_values[ecx*8]
; Line 187
    sub    esp, 8
    fld    QWORD PTR _x$[ebp]
    fstp   QWORD PTR [esp]
    call   _log
    add    esp, 8
    mov    edx, DWORD PTR _numval
    fstp   QWORD PTR _log_x[edx*8]
; Line 188
    sub    esp, 8
    fld    QWORD PTR _y$[ebp]
    fstp   QWORD PTR [esp]
    call   _log
    add    esp, 8
    mov    eax, DWORD PTR _numval
    fstp   QWORD PTR _log_y[eax*8]
; Line 190
    mov    ecx, DWORD PTR _numval
    add    ecx, 1
    mov    DWORD PTR numval, ecx
; Line 191
    jmp    $LN15@setxyvals
$LN9@setxyvals:
; Line 192
    mov    ecx, DWORD PTR ___$ArrayPad$[ebp]
    xor    ecx, ebp
    call   @__security_check_cookie@4
    mov    esp, ebp
    pop    ebp
    ret    0
_setxyvals ENDP
_TEXT ENDS
PUBLIC __real@3fb9999999999999a
PUBLIC _test_predictions
EXTRN _pow:PROC
EXTRN _exp:PROC
; COMDAT __real@3fb9999999999999a
CONST SEGMENT
    real@3fb9999999999999a DQ 03fb999999999999ar ; 0.1
; Function compile flags: /Odtp
CONST ENDS
_TEXT SEGMENT
tv72 = -60 ; size = 4
_y_actual$ = -56 ; size = 8
_err$ = -48 ; size = 8
```

```
_tot_err$ = -40 ; size = 8
_i$ = -28 ; size = 4
_y_diff$ = -24 ; size = 8
_x$ = -16 ; size = 8
_y_pred$ = -8 ; size = 8
_numval$ = 8 ; size = 4
_x_values$ = 12 ; size = 4
_y_values$ = 16 ; size = 4
_save_type$ = 20 ; size = 4
_save_a1$ = 24 ; size = 8
_save_b1$ = 32 ; size = 8
_test_predictions PROC
; Line 199
    push    ebp
    mov     ebp, esp
    sub     esp, 60 ; 0000003cH
; Line 203
    fldz
    fstp    QWORD PTR _tot_err$[ebp]
; Line 205
    fld     QWORD PTR _save_a1$[ebp]
    fadd   QWORD PTR _a1_weight
    fstp    QWORD PTR _save_a1$[ebp]
; Line 206
    fld     QWORD PTR _save_b1$[ebp]
    fadd   QWORD PTR _b1_weight
    fstp    QWORD PTR _save_b1$[ebp]
; Line 208
    mov     DWORD PTR _i$[ebp], 0
    jmp     SHORT $LN14@test_predi
$LN13@test_predi:
    mov     eax, DWORD PTR _i$[ebp]
    add     eax, 1
    mov     DWORD PTR i$[ebp], eax
$LN14@test_predi:
    mov     ecx, DWORD PTR _i$[ebp]
    cmp     ecx, DWORD PTR _numval$[ebp]
    jge     $LN12@test_predi
; Line 210
    mov     edx, DWORD PTR i$[ebp]
    mov     eax, DWORD PTR _x_values$[ebp]
    fld     QWORD PTR [eax+edx*8]
    fstp    QWORD PTR _x$[ebp]
; Line 211
    mov     ecx, DWORD PTR _i$[ebp]
    mov     edx, DWORD PTR _y_values$[ebp]
    fld     QWORD PTR [edx+ecx*8]
    fstp    QWORD PTR _y_actual$[ebp]
; Line 212
    mov     eax, DWORD PTR _save_type$[ebp]
    mov     DWORD PTR tv72[ebp], eax
    mov     ecx, DWORD PTR tv72[ebp]
    sub     ecx, 1
    mov     DWORD PTR tv72[ebp], ecx
    cmp     DWORD PTR tv72[ebp], 3
    ja     $LN10@test_predi
    mov     edx, DWORD PTR tv72[ebp]
    jmp     DWORD PTR $LN17@test_predi[edx*4]
$LN9@test_predi:
; Line 214
    fld     QWORD PTR _save_b1$[ebp]
    fmul   QWORD PTR _x$[ebp]
    fadd   QWORD PTR _save_a1$[ebp]
    fstp    QWORD PTR _y_pred$[ebp]
    jmp     $LN10@test_predi
```

\$LN8@test_predi:

```
; Line 215
    sub     esp, 8
    fld     QWORD PTR _x$[ebp]
    fstp    QWORD PTR [esp]
    call    _log
    add     esp, 8
    fmul    QWORD PTR _save_b1$[ebp]
    fadd    QWORD PTR _save_a1$[ebp]
    fstp    QWORD PTR _y_pred$[ebp]
    jmp     $LN10@test_predi
```

\$LN7@test_predi:

```
; Line 216
    sub     esp, 8
    fld     QWORD PTR _save_a1$[ebp]
    fstp    QWORD PTR [esp]
    call    _exp
    add     esp, 8
    fstp    QWORD PTR _save_a1$[ebp]
    fld     QWORD PTR _save_a1$[ebp]
    fmul    QWORD PTR _save_b1$[ebp]
    sub     esp, 8
    fstp    QWORD PTR [esp]
    call    _exp
    add     esp, 8
    fmul    QWORD PTR _save_a1$[ebp]
    fstp    QWORD PTR _y_pred$[ebp]
    jmp     SHORT $LN10@test_predi
```

\$LN6@test_predi:

```
; Line 217
    sub     esp, 8
    fld     QWORD PTR _save_b1$[ebp]
    fstp    QWORD PTR [esp]
    sub     esp, 8
    fld     QWORD PTR _x$[ebp]
    fstp    QWORD PTR [esp]
    call    _pow
    add     esp, 16
    fmul    QWORD PTR _save_a1$[ebp]
    fstp    QWORD PTR _y_pred$[ebp]
    sub     esp, 8
    fld     QWORD PTR _save_b1$[ebp]
    fstp    QWORD PTR [esp]
    sub     esp, 8
    fld     QWORD PTR _x$[ebp]
    fstp    QWORD PTR [esp]
    sub     esp, 8
    fld     QWORD PTR _save_a1$[ebp]
    fstp    QWORD PTR [esp]
    sub     esp, 8
    fld     QWORD PTR _y_pred$[ebp]
    fstp    QWORD PTR [esp]
    push    OFFSET $SG5211
    call    _printf
    add     esp, 36
; 00000010H
; 00000024H
```

\$LN10@test_predi:

```
; Line 220
    fld     QWORD PTR _y_actual$[ebp]
    fsub    QWORD PTR _y_pred$[ebp]
    fstp    QWORD PTR _y_diff$[ebp]
; Line 221
    fld     QWORD PTR _y_diff$[ebp]
    fdiv    QWORD PTR _y_actual$[ebp]
    fstp    QWORD PTR _err$[ebp]
; Line 222
```

```
    cmp     DWORD PTR _print_row, 0
    je      SHORT $LN5@test_predi
; Line 223
    sub     esp, 8
    fld     QWORD PTR _err$[ebp]
    fstp    QWORD PTR [esp]
    sub     esp, 8
    fld     QWORD PTR _y_diff$[ebp]
    fstp    QWORD PTR [esp]
    sub     esp, 8
    fld     QWORD PTR _y_pred$[ebp]
    fstp    QWORD PTR [esp]
    sub     esp, 8
    fld     QWORD PTR _y_actual$[ebp]
    fstp    QWORD PTR [esp]
    sub     esp, 8
    fld     QWORD PTR _x$[ebp]
    fstp    QWORD PTR [esp]
    push   OFFSET $SG5213
    call    _printf
    add     esp, 44 ; 0000002cH
$LN5@test_predi:
; Line 224
    fld     QWORD PTR _tot_err$[ebp]
    fadd    QWORD PTR _err$[ebp]
    fstp    QWORD PTR _tot_err$[ebp]
; Line 225
    jmp     $LN13@test_predi
$LN12@test_predi:
; Line 227
    sub     esp, 8
    fld     QWORD PTR _tot_err$[ebp]
    fstp    QWORD PTR [esp]
    sub     esp, 8
    fld     QWORD PTR _b1_weight
    fstp    QWORD PTR [esp]
    sub     esp, 8
    fld     QWORD PTR _a1_weight
    fstp    QWORD PTR [esp]
    sub     esp, 8
    fld     QWORD PTR _save_b1$[ebp]
    fstp    QWORD PTR [esp]
    sub     esp, 8
    fld     QWORD PTR _save_a1$[ebp]
    fstp    QWORD PTR [esp]
    push   OFFSET $SG5214
    call    _printf
    add     esp, 44 ; 0000002cH
; Line 229
    fldz
    fcomp   QWORD PTR _tot_err$[ebp]
    fnstsw ax
    test    ah, 65 ; 00000041H
    jne     SHORT $LN4@test_predi
; Line 231
    fld     QWORD PTR _a1_weight
    fadd    QWORD PTR __real@3fb9999999999999a
    fstp    QWORD PTR _a1_weight
; Line 232
    fld     QWORD PTR _b1_weight
    fsub    QWORD PTR __real@3fb9999999999999a
    fstp    QWORD PTR _b1_weight
    jmp     SHORT $LN1@test_predi
$LN4@test_predi:
; Line 234
```



```
fldz
fcomp QWORD PTR _tot_err$[ebp]
fnstsw ax
test ah, 5
jp SHORT $LN2@test_predi
; Line 236
fld QWORD PTR _a1_weight
fsub QWORD PTR ___real@3fb9999999999999a
fstp QWORD PTR a1_weight
; Line 237
fld QWORD PTR _b1_weight
fadd QWORD PTR ___real@3fb9999999999999a
fstp QWORD PTR b1_weight
jmp SHORT $LN1@test_predi
$LN2@test_predi:
; Line 239
push OFFSET $SG5219
call _puts
add esp, 4
push 0
call _exit
$LN1@test_predi:
; Line 240
mov esp, ebp
pop ebp
ret 0
$LN17@test_predi:
DD $LN6@test_predi
DD $LN8@test_predi
DD $LN7@test_predi
DD $LN9@test_predi
_test_predictions ENDP
_TEXT ENDS
PUBLIC real@40026bb1bbb55519
PUBLIC ___real@3ff028f5c28f5c29
PUBLIC _main
; COMDAT ___real@40026bb1bbb55519
CONST SEGMENT
__real@40026bb1bbb55519 DQ 040026bb1bbb55519r ; 2.30259
CONST ENDS
; COMDAT ___real@3ff028f5c28f5c29
CONST SEGMENT
__real@3ff028f5c28f5c29 DQ 03ff028f5c28f5c29r ; 1.01
; Function compile flags: /Odtp
CONST ENDS
_TEXT SEGMENT
tv131 = -60 ; size = 4
_d$5238 = -56 ; size = 8
_i$5239 = -44 ; size = 4
_save_a1$ = -40 ; size = 8
_save_type$ = -28 ; size = 4
_save_r2$ = -24 ; size = 8
_save_b1$ = -16 ; size = 8
_i$ = -4 ; size = 4
_argc$ = 8 ; size = 4
_argv$ = 12 ; size = 4
_main PROC
; Line 246
push ebp
mov ebp, esp
sub esp, 60 ; 0000003cH
; Line 248
mov DWORD PTR _save_type$[ebp], -1
; Line 250
call _setxyvals
```

```
; Line 251
    fldz
    fstp    QWORD PTR _save_r2$[ebp]
; Line 253
    mov     DWORD PTR _i$[ebp], 1
    jmp     SHORT $LN22@main
$LN21@main:
    mov     eax, DWORD PTR _i$[ebp]
    add     eax, 1
    mov     DWORD PTR _i$[ebp], eax
$LN22@main:
    cmp     DWORD PTR _i$[ebp], 4
    jg     $LN20@main
; Line 255
    mov     ecx, DWORD PTR numval
    push    ecx
    push    OFFSET _y_values
    push    OFFSET _x_values
    mov     edx, DWORD PTR _i$[ebp]
    push    edx
    call    _do_regress
; Line 257
    add     esp, 8
    fld     QWORD PTR _r2
    fstp    QWORD PTR [esp]
    sub     esp, 8
    fld     QWORD PTR _b1
    fstp    QWORD PTR [esp]
    sub     esp, 8
    fld     QWORD PTR _a1
    fstp    QWORD PTR [esp]
    mov     eax, DWORD PTR _i$[ebp]
    mov     ecx, DWORD PTR _regress_name[eax*4]
    push    ecx
    push    OFFSET $SG5234
    call    _printf
    add     esp, 32                                ; 00000020H
; Line 258
    fldz
    fcomp   QWORD PTR r2
    fnstsw ax
    test    ah, 5
    jp     SHORT $LN18@main
    fld     QWORD PTR __real@3ff028f5c28f5c29
    fcomp   QWORD PTR _r2
    fnstsw ax
    test    ah, 1
    jne    SHORT $LN18@main
; Line 259
    fld     QWORD PTR _save_r2$[ebp]
    fcomp   QWORD PTR _r2
    fnstsw ax
    test    ah, 5
    jp     SHORT $LN18@main
; Line 261
    fld     QWORD PTR _r2
    fstp    QWORD PTR _save_r2$[ebp]
; Line 262
    mov     edx, DWORD PTR i$[ebp]
    mov     DWORD PTR _save_type$[ebp], edx
; Line 263
    fld     QWORD PTR _a1
    fstp    QWORD PTR _save_a1$[ebp]
; Line 264
    fld     QWORD PTR _b1
```

```
        fstp     QWORD PTR _save_b1$[ebp]
$LN18@main:
; Line 266
        jmp     $LN21@main
$LN20@main:
; Line 268
        cmp     DWORD PTR _save_type$[ebp], -1
        jne     $LN17@main
; Line 272
        push   OFFSET $SG5240
        call   _printf
        add     esp, 4
; Line 274
        fld     QWORD PTR _y_values
        fstp   QWORD PTR _d$5238[ebp]
; Line 275
        mov     DWORD PTR _i$5239[ebp], 1
        jmp     $LN16@main
$LN15@main:
        mov     eax, DWORD PTR _i$5239[ebp]
        add     eax, 1
        mov     DWORD PTR _i$5239[ebp], eax
$LN16@main:
        mov     ecx, DWORD PTR _i$5239[ebp]
        cmp     ecx, DWORD PTR _numval
        jge     $LN14@main
; Line 276
        mov     edx, DWORD PTR _i$5239[ebp]
        fld     QWORD PTR _y_values[edx*8]
        fcomp  QWORD PTR _d$5238[ebp]
        fnstsw ax
        test   ah, 68 ; 00000044H
        jnp    $LN13@main
; Line 277
        jmp     $LN14@main
$LN13@main:
; Line 278
        jmp     $LN15@main
$LN14@main:
        mov     eax, DWORD PTR _i$5239[ebp]
        cmp     eax, DWORD PTR _numval
        jne     $LN12@main
; Line 279
        sub     esp, 8
        fld     QWORD PTR _d$5238[ebp]
        fstp   QWORD PTR [esp]
        push   OFFSET $SG5246
        call   _printf
        add     esp, 12 ; 0000000cH
$LN12@main:
; Line 281
        fld     QWORD PTR _x_values
        fstp   QWORD PTR _d$5238[ebp]
; Line 282
        mov     DWORD PTR _i$5239[ebp], 1
        jmp     $LN11@main
$LN10@main:
        mov     ecx, DWORD PTR _i$5239[ebp]
        add     ecx, 1
        mov     DWORD PTR _i$5239[ebp], ecx
$LN11@main:
        mov     edx, DWORD PTR _i$5239[ebp]
        cmp     edx, DWORD PTR _numval
        jge     $LN9@main
; Line 283
```

```
    mov     eax, DWORD PTR _i$5239[ebp]
    fld     QWORD PTR _x_values[eax*8]
    fcomp  QWORD PTR _d$5238[ebp]
    fnstsw ax
    test   ah, 68 ; 00000044H
    jnp   SHORT $LN8@main
; Line 284
    jmp   SHORT $LN9@main
$LN8@main:
; Line 285
    jmp   SHORT $LN10@main
$LN9@main:
    mov     ecx, DWORD PTR _i$5239[ebp]
    cmp     ecx, DWORD PTR _numval
    jne    SHORT $LN7@main
; Line 286
    sub     esp, 8
    fld     QWORD PTR _d$5238[ebp]
    fstp   QWORD PTR [esp]
    push   OFFSET $SG5252
    call   _printf
    add     esp, 12 ; 0000000cH
$LN7@main:
; Line 287
    push   0
    call   _exit
$LN17@main:
; Line 290
    sub     esp, 8
    fld     QWORD PTR _save_r2$[ebp]
    fstp   QWORD PTR [esp]
    push   OFFSET $SG5253
    call   _printf
    add     esp, 12 ; 0000000cH
; Line 292
    mov     edx, DWORD PTR _save_type$[ebp]
    mov     eax, DWORD PTR _regress_name[edx*4]
    push   eax
    push   OFFSET $SG5254
    call   printf
    add     esp, 8
; Line 294
    mov     ecx, DWORD PTR _save_type$[ebp]
    mov     DWORD PTR tv131[ebp], ecx
    mov     edx, DWORD PTR tv131[ebp]
    sub     edx, 1
    mov     DWORD PTR tv131[ebp], edx
    cmp     DWORD PTR tv131[ebp], 3
    ja     $LN5@main
    mov     eax, DWORD PTR tv131[ebp]
    jmp    DWORD PTR $LN27@main[eax*4]
$LN4@main:
; Line 297
    sub     esp, 8
    fld     QWORD PTR _save_b1$[ebp]
    fstp   QWORD PTR [esp]
    sub     esp, 8
    fld     QWORD PTR _save_a1$[ebp]
    fstp   QWORD PTR [esp]
    push   OFFSET $SG5260
    call   _printf
; Line 298
    add     esp, 12 ; 0000000cH
    fld     QWORD PTR _save_b1$[ebp]
    fstp   QWORD PTR [esp]
```

```
    sub     esp, 8
    fld     QWORD PTR _save_a1$[ebp]
    fstp    QWORD PTR [esp]
    push    OFFSET $SG5261
    call    _printf
    add     esp, 20                                ; 00000014H
; Line 299
    jmp     $LN5@main
$LN3@main:
; Line 301
    sub     esp, 8
    fld     QWORD PTR _save_b1$[ebp]
    fstp    QWORD PTR [esp]
    sub     esp, 8
    fld     QWORD PTR save_a1$[ebp]
    fstp    QWORD PTR [esp]
    push    OFFSET $SG5263
    call    _printf
; Line 302
    add     esp, 12                                ; 0000000cH
    fld     QWORD PTR _save_b1$[ebp]
    fstp    QWORD PTR [esp]
    sub     esp, 8
    fld     QWORD PTR _save_a1$[ebp]
    fstp    QWORD PTR [esp]
    push    OFFSET $SG5264
    call    _printf
    add     esp, 20                                ; 00000014H
; Line 303
    fld     QWORD PTR _save_b1$[ebp]
    fmul    QWORD PTR __real@40026bb1bbb55519
    fstp    QWORD PTR _save_b1$[ebp]
; Line 304
    sub     esp, 8
    fld     QWORD PTR _save_b1$[ebp]
    fstp    QWORD PTR [esp]
    sub     esp, 8
    fld     QWORD PTR _save_a1$[ebp]
    fstp    QWORD PTR [esp]
    push    OFFSET $SG5265
    call    _printf
    add     esp, 20                                ; 00000014H
; Line 305
    jmp     $LN5@main
$LN2@main:
; Line 307
    sub     esp, 8
    fld     QWORD PTR _save_a1$[ebp]
    fstp    QWORD PTR [esp]
    call    _exp
    add     esp, 8
    fstp    QWORD PTR _save_a1$[ebp]
; Line 308
    sub     esp, 8
    fld     QWORD PTR _save_b1$[ebp]
    fstp    QWORD PTR [esp]
    sub     esp, 8
    fld     QWORD PTR _save_a1$[ebp]
    fstp    QWORD PTR [esp]
    push    OFFSET $SG5267
    call    _printf
; Line 309
    add     esp, 12                                ; 0000000cH
    fld     QWORD PTR _save_b1$[ebp]
    fstp    QWORD PTR [esp]
```

```
    sub     esp, 8
    fld     QWORD PTR _save_a1$[ebp]
    fstp   QWORD PTR [esp]
    push   OFFSET $SG5268
    call   _printf
    add    esp, 20                                ; 00000014H
; Line 310
    jmp    SHORT $LN5@main
$LN1@main:
; Line 312
    sub     esp, 8
    fld     QWORD PTR _save_a1$[ebp]
    fstp   QWORD PTR [esp]
    call   _exp
    add    esp, 8
    fstp   QWORD PTR _save_a1$[ebp]
; Line 313
    sub     esp, 8
    fld     QWORD PTR _save_b1$[ebp]
    fstp   QWORD PTR [esp]
    sub    esp, 8
    fld     QWORD PTR _save_a1$[ebp]
    fstp   QWORD PTR [esp]
    push   OFFSET $SG5270
    call   _printf
; Line 314
    add    esp, 12                                ; 0000000cH
    fld     QWORD PTR _save_b1$[ebp]
    fstp   QWORD PTR [esp]
    sub    esp, 8
    fld     QWORD PTR _save_a1$[ebp]
    fstp   QWORD PTR [esp]
    push   OFFSET $SG5271
    call   printf
    add    esp, 20                                ; 00000014H
$LN5@main:
; Line 317
    push   OFFSET $SG5272
    call   _printf
    add    esp, 4
; Line 331
    jmp    SHORT $LN26@main
$LN24@main:
    jmp    SHORT $LN25@main
$LN26@main:
    xor    eax, eax
$LN25@main:
    mov    esp, ebp
    pop    ebp
    ret    0
$LN27@main:
    DD    $LN1@main
    DD    $LN3@main
    DD    $LN2@main
    DD    $LN4@main
_main   ENDP
_TEXT  ENDS
END
```