```
Intel(R) Advisor can now assist with vectorization and show optimization
  report messages with your source code.
See "https://software.intel.com/en-us/intel-advisor-xe" for details.


Begin optimization report for: main(int, char **)

    Report from: Vector optimizations [vec]


LOOP BEGIN at stencil.c(76,5) inlined into stencil.c(40,3)
   remark #15542: loop was not vectorized: inner loop was already vectorized

   LOOP BEGIN at stencil.c(75,3) inlined into stencil.c(40,3)
   <Peeled loop for vectorization>
   LOOP END

   LOOP BEGIN at stencil.c(75,3) inlined into stencil.c(40,3)
      remark #15389: vectorization support: reference image has unaligned access
[ stencil.c(77,7) ]
      remark #15388: vectorization support: reference tmp_image has aligned
access   [ stencil.c(78,7) ]
      remark #15381: vectorization support: unaligned access used inside loop body
      remark #15305: vectorization support: vector length 8
      remark #15399: vectorization support: unroll factor set to 2
      remark #15309: vectorization support: normalized vectorization overhead 0.464
      remark #15301: PERMUTED LOOP WAS VECTORIZED
      remark #15442: entire loop may be executed in remainder
      remark #15449: unmasked aligned unit stride stores: 1
      remark #15451: unmasked unaligned unit stride stores: 1
      remark #15475: --- begin vector loop cost summary ---
      remark #15476: scalar loop cost: 13
      remark #15477: vector loop cost: 1.750
      remark #15478: estimated potential speedup: 5.790
      remark #15488: --- end vector loop cost summary ---
   LOOP END

   LOOP BEGIN at stencil.c(75,3) inlined into stencil.c(40,3)
   <Remainder loop for vectorization>
      remark #15389: vectorization support: reference image has unaligned access
[ stencil.c(77,7) ]
      remark #15388: vectorization support: reference tmp_image has aligned
access   [ stencil.c(78,7) ]
      remark #15381: vectorization support: unaligned access used inside loop body
      remark #15305: vectorization support: vector length 4
      remark #15309: vectorization support: normalized vectorization overhead 1.444
      remark #15301: REMAINDER LOOP WAS VECTORIZED
   LOOP END

   LOOP BEGIN at stencil.c(75,3) inlined into stencil.c(40,3)
   <Remainder loop for vectorization>
   LOOP END
LOOP END

LOOP BEGIN at stencil.c(83,3) inlined into stencil.c(40,3)
   remark #15542: loop was not vectorized: inner loop was already vectorized

   LOOP BEGIN at stencil.c(84,5) inlined into stencil.c(40,3)
      remark #15542: loop was not vectorized: inner loop was already vectorized

      LOOP BEGIN at stencil.c(85,7) inlined into stencil.c(40,3)
         remark #15542: loop was not vectorized: inner loop was already vectorized

         LOOP BEGIN at stencil.c(86,9) inlined into stencil.c(40,3)
            remark #15329: vectorization support: scatter was emulated for the
variable image:  strided by non-constant value   [ stencil.c(88,11) ]
```

```
            remark #15305: vectorization support: vector length 16
            remark #15300: LOOP WAS VECTORIZED
            remark #15462: unmasked indexed (or gather) loads: 1
            remark #15475: --- begin vector loop cost summary ---
            remark #15476: scalar loop cost: 6
            remark #15477: vector loop cost: 2.430
            remark #15478: estimated potential speedup: 2.320
            remark #15488: --- end vector loop cost summary ---
         LOOP END

         LOOP BEGIN at stencil.c(86,9) inlined into stencil.c(40,3)
         <Remainder loop for vectorization>
            remark #15305: vectorization support: vector length 4
            remark #15309: vectorization support: normalized vectorization
overhead 0.833
            remark #15301: REMAINDER LOOP WAS VECTORIZED
         LOOP END

         LOOP BEGIN at stencil.c(86,9) inlined into stencil.c(40,3)
         <Remainder loop for vectorization>
         LOOP END
      LOOP END
   LOOP END
LOOP END

LOOP BEGIN at stencil.c(44,3)
   remark #15542: loop was not vectorized: inner loop was already vectorized

   LOOP BEGIN at stencil.c(61,3) inlined into stencil.c(45,5)
      remark #15542: loop was not vectorized: inner loop was already vectorized

      LOOP BEGIN at stencil.c(62,5) inlined into stencil.c(45,5)
      <Peeled loop for vectorization>
      LOOP END

      LOOP BEGIN at stencil.c(62,5) inlined into stencil.c(45,5)
         remark #15388: vectorization support: reference tmp_image has aligned
access   [ stencil.c(63,7) ]
         remark #15389: vectorization support: reference image has unaligned
access   [ stencil.c(63,7) ]
         remark #15388: vectorization support: reference tmp_image has aligned
access   [ stencil.c(64,21) ]
         remark #15388: vectorization support: reference tmp_image has aligned
access   [ stencil.c(64,21) ]
         remark #15389: vectorization support: reference image has unaligned
access   [ stencil.c(64,21) ]
         remark #15388: vectorization support: reference tmp_image has aligned
access   [ stencil.c(65,21) ]
         remark #15388: vectorization support: reference tmp_image has aligned
access   [ stencil.c(65,21) ]
         remark #15389: vectorization support: reference image has unaligned
access   [ stencil.c(65,21) ]
         remark #15388: vectorization support: reference tmp_image has aligned
access   [ stencil.c(66,21) ]
         remark #15388: vectorization support: reference tmp_image has aligned
access   [ stencil.c(66,21) ]
         remark #15389: vectorization support: reference image has unaligned
access   [ stencil.c(66,21) ]
         remark #15388: vectorization support: reference tmp_image has aligned
access   [ stencil.c(67,21) ]
         remark #15388: vectorization support: reference tmp_image has aligned
access   [ stencil.c(67,21) ]
         remark #15389: vectorization support: reference image has unaligned
access   [ stencil.c(67,21) ]
         remark #15381: vectorization support: unaligned access used inside loop
body
```

```
         remark #15305: vectorization support: vector length 8
         remark #15309: vectorization support: normalized vectorization overhead
0.405
         remark #15300: LOOP WAS VECTORIZED
         remark #15442: entire loop may be executed in remainder
         remark #15448: unmasked aligned unit stride loads: 1
         remark #15449: unmasked aligned unit stride stores: 4
         remark #15450: unmasked unaligned unit stride loads: 4
         remark #15455: masked aligned unit stride stores: 1
         remark #15456: masked unaligned unit stride loads: 1
         remark #15475: --- begin vector loop cost summary ---
         remark #15476: scalar loop cost: 97
         remark #15477: vector loop cost: 9.870
         remark #15478: estimated potential speedup: 7.200
         remark #15488: --- end vector loop cost summary ---
      LOOP END

      LOOP BEGIN at stencil.c(62,5) inlined into stencil.c(45,5)
      <Remainder loop for vectorization>
      LOOP END

      LOOP BEGIN at stencil.c(62,5) inlined into stencil.c(45,5)
      <Peeled loop for vectorization>
      LOOP END

      LOOP BEGIN at stencil.c(62,5) inlined into stencil.c(45,5)
         remark #15388: vectorization support: reference tmp_image has aligned
access   [ stencil.c(63,7) ]
         remark #15389: vectorization support: reference image has unaligned
access   [ stencil.c(63,7) ]
         remark #15388: vectorization support: reference tmp_image has aligned
access   [ stencil.c(64,21) ]
         remark #15388: vectorization support: reference tmp_image has aligned
access   [ stencil.c(64,21) ]
         remark #15389: vectorization support: reference image has unaligned
access   [ stencil.c(64,21) ]
         remark #15388: vectorization support: reference tmp_image has aligned
access   [ stencil.c(66,21) ]
         remark #15388: vectorization support: reference tmp_image has aligned
access   [ stencil.c(66,21) ]
         remark #15389: vectorization support: reference image has unaligned
access   [ stencil.c(66,21) ]
         remark #15388: vectorization support: reference tmp_image has aligned
access   [ stencil.c(67,21) ]
         remark #15388: vectorization support: reference tmp_image has aligned
access   [ stencil.c(67,21) ]
         remark #15389: vectorization support: reference image has unaligned
access   [ stencil.c(67,21) ]
         remark #15381: vectorization support: unaligned access used inside loop
body
         remark #15305: vectorization support: vector length 8
         remark #15309: vectorization support: normalized vectorization overhead
0.439
         remark #15300: LOOP WAS VECTORIZED
         remark #15442: entire loop may be executed in remainder
         remark #15448: unmasked aligned unit stride loads: 1
         remark #15449: unmasked aligned unit stride stores: 3
         remark #15450: unmasked unaligned unit stride loads: 3
         remark #15455: masked aligned unit stride stores: 1
         remark #15456: masked unaligned unit stride loads: 1
         remark #15475: --- begin vector loop cost summary ---
         remark #15476: scalar loop cost: 78
         remark #15477: vector loop cost: 8.250
         remark #15478: estimated potential speedup: 7.000
         remark #15488: --- end vector loop cost summary ---
      LOOP END
      LOOP BEGIN at stencil.c(62,5) inlined into stencil.c(45,5)
      <Remainder loop for vectorization>
      LOOP END

      LOOP BEGIN at stencil.c(62,5) inlined into stencil.c(45,5)
      <Peeled loop for vectorization>
      LOOP END

      LOOP BEGIN at stencil.c(62,5) inlined into stencil.c(45,5)
         remark #15388: vectorization support: reference tmp_image has aligned
access   [ stencil.c(63,7) ]
         remark #15389: vectorization support: reference image has unaligned
access   [ stencil.c(63,7) ]
         remark #15388: vectorization support: reference tmp_image has aligned
access   [ stencil.c(65,21) ]
         remark #15388: vectorization support: reference tmp_image has aligned
access   [ stencil.c(65,21) ]
         remark #15389: vectorization support: reference image has unaligned
access   [ stencil.c(65,21) ]
         remark #15388: vectorization support: reference tmp_image has aligned
access   [ stencil.c(66,21) ]
         remark #15388: vectorization support: reference tmp_image has aligned
access   [ stencil.c(66,21) ]
         remark #15389: vectorization support: reference image has unaligned
access   [ stencil.c(66,21) ]
         remark #15388: vectorization support: reference tmp_image has aligned
access   [ stencil.c(67,21) ]
         remark #15388: vectorization support: reference tmp_image has aligned
access   [ stencil.c(67,21) ]
         remark #15389: vectorization support: reference image has unaligned
access   [ stencil.c(67,21) ]
         remark #15381: vectorization support: unaligned access used inside loop
body
         remark #15305: vectorization support: vector length 8
         remark #15309: vectorization support: normalized vectorization overhead
0.439
         remark #15300: LOOP WAS VECTORIZED
         remark #15442: entire loop may be executed in remainder
         remark #15448: unmasked aligned unit stride loads: 1
         remark #15449: unmasked aligned unit stride stores: 3
         remark #15450: unmasked unaligned unit stride loads: 3
         remark #15455: masked aligned unit stride stores: 1
         remark #15456: masked unaligned unit stride loads: 1
         remark #15475: --- begin vector loop cost summary ---
         remark #15476: scalar loop cost: 78
         remark #15477: vector loop cost: 8.250
         remark #15478: estimated potential speedup: 7.000
         remark #15488: --- end vector loop cost summary ---
      LOOP END

      LOOP BEGIN at stencil.c(62,5) inlined into stencil.c(45,5)
      <Remainder loop for vectorization>
      LOOP END

      LOOP BEGIN at stencil.c(62,5) inlined into stencil.c(45,5)
      <Peeled loop for vectorization>
      LOOP END

      LOOP BEGIN at stencil.c(62,5) inlined into stencil.c(45,5)
         remark #15388: vectorization support: reference tmp_image has aligned
access   [ stencil.c(63,7) ]
         remark #15389: vectorization support: reference image has unaligned
access   [ stencil.c(63,7) ]
         remark #15388: vectorization support: reference tmp_image has aligned
access   [ stencil.c(66,21) ]
```

```
         remark #15388: vectorization support: reference tmp_image has aligned 
access   [ stencil.c(66,21) ]
         remark #15389: vectorization support: reference image has unaligned 
access   [ stencil.c(66,21) ]
         remark #15388: vectorization support: reference tmp_image has aligned 
access   [ stencil.c(67,21) ]
         remark #15388: vectorization support: reference tmp_image has aligned 
access   [ stencil.c(67,21) ]
         remark #15389: vectorization support: reference image has unaligned 
access   [ stencil.c(67,21) ]
         remark #15381: vectorization support: unaligned access used inside loop 
body
         remark #15305: vectorization support: vector length 4
         remark #15309: vectorization support: normalized vectorization overhead 
0.719
         remark #15300: LOOP WAS VECTORIZED
         remark #15442: entire loop may be executed in remainder
         remark #15448: unmasked aligned unit stride loads: 1 
         remark #15449: unmasked aligned unit stride stores: 2 
         remark #15450: unmasked unaligned unit stride loads: 2 
         remark #15455: masked aligned unit stride stores: 1 
         remark #15456: masked unaligned unit stride loads: 1 
         remark #15475: --- begin vector loop cost summary ---
         remark #15476: scalar loop cost: 58 
         remark #15477: vector loop cost: 8.000 
         remark #15478: estimated potential speedup: 6.630 
         remark #15488: --- end vector loop cost summary ---
      LOOP END

      LOOP BEGIN at stencil.c(62,5) inlined into stencil.c(45,5)
      <Remainder loop for vectorization>
      LOOP END
   LOOP END

   LOOP BEGIN at stencil.c(61,3) inlined into stencil.c(46,5)
      remark #15542: loop was not vectorized: inner loop was already vectorized

      LOOP BEGIN at stencil.c(62,5) inlined into stencil.c(46,5)
      <Peeled loop for vectorization>
      LOOP END

      LOOP BEGIN at stencil.c(62,5) inlined into stencil.c(46,5)
         remark #15388: vectorization support: reference tmp_image has aligned 
access   [ stencil.c(63,7) ]
         remark #15389: vectorization support: reference image has unaligned 
access   [ stencil.c(63,7) ]
         remark #15388: vectorization support: reference tmp_image has aligned 
access   [ stencil.c(64,21) ]
         remark #15388: vectorization support: reference tmp_image has aligned 
access   [ stencil.c(64,21) ]
         remark #15389: vectorization support: reference image has unaligned 
access   [ stencil.c(64,21) ]
         remark #15388: vectorization support: reference tmp_image has aligned 
access   [ stencil.c(65,21) ]
         remark #15388: vectorization support: reference tmp_image has aligned 
access   [ stencil.c(65,21) ]
         remark #15389: vectorization support: reference image has unaligned 
access   [ stencil.c(65,21) ]
         remark #15388: vectorization support: reference tmp_image has aligned 
access   [ stencil.c(66,21) ]
         remark #15388: vectorization support: reference tmp_image has aligned 
access   [ stencil.c(66,21) ]
         remark #15389: vectorization support: reference image has unaligned 
access   [ stencil.c(66,21) ]
         remark #15388: vectorization support: reference tmp_image has aligned 
access   [ stencil.c(67,21) ]
```

```
         remark #15388: vectorization support: reference tmp_image has aligned
access   [ stencil.c(67,21) ]
         remark #15389: vectorization support: reference image has unaligned
access   [ stencil.c(67,21) ]
         remark #15381: vectorization support: unaligned access used inside loop
body
         remark #15305: vectorization support: vector length 8
         remark #15309: vectorization support: normalized vectorization overhead
0.405
         remark #15300: LOOP WAS VECTORIZED
         remark #15442: entire loop may be executed in remainder
         remark #15448: unmasked aligned unit stride loads: 1
         remark #15449: unmasked aligned unit stride stores: 4
         remark #15450: unmasked unaligned unit stride loads: 4
         remark #15455: masked aligned unit stride stores: 1
         remark #15456: masked unaligned unit stride loads: 1
         remark #15475: --- begin vector loop cost summary ---
         remark #15476: scalar loop cost: 97
         remark #15477: vector loop cost: 9.870
         remark #15478: estimated potential speedup: 7.200
         remark #15488: --- end vector loop cost summary ---
      LOOP END

      LOOP BEGIN at stencil.c(62,5) inlined into stencil.c(46,5)
      <Remainder loop for vectorization>
      LOOP END

      LOOP BEGIN at stencil.c(62,5) inlined into stencil.c(46,5)
      <Peeled loop for vectorization>
      LOOP END

      LOOP BEGIN at stencil.c(62,5) inlined into stencil.c(46,5)
         remark #15388: vectorization support: reference tmp_image has aligned
access   [ stencil.c(63,7) ]
         remark #15389: vectorization support: reference image has unaligned
access   [ stencil.c(63,7) ]
         remark #15388: vectorization support: reference tmp_image has aligned
access   [ stencil.c(64,21) ]
         remark #15388: vectorization support: reference tmp_image has aligned
access   [ stencil.c(64,21) ]
         remark #15389: vectorization support: reference image has unaligned
access   [ stencil.c(64,21) ]
         remark #15388: vectorization support: reference tmp_image has aligned
access   [ stencil.c(66,21) ]
         remark #15388: vectorization support: reference tmp_image has aligned
access   [ stencil.c(66,21) ]
         remark #15389: vectorization support: reference image has unaligned
access   [ stencil.c(66,21) ]
         remark #15388: vectorization support: reference tmp_image has aligned
access   [ stencil.c(67,21) ]
         remark #15388: vectorization support: reference tmp_image has aligned
access   [ stencil.c(67,21) ]
         remark #15389: vectorization support: reference image has unaligned
access   [ stencil.c(67,21) ]
         remark #15381: vectorization support: unaligned access used inside loop
body
         remark #15305: vectorization support: vector length 8
         remark #15309: vectorization support: normalized vectorization overhead
0.439
         remark #15300: LOOP WAS VECTORIZED
         remark #15442: entire loop may be executed in remainder
         remark #15448: unmasked aligned unit stride loads: 1
         remark #15449: unmasked aligned unit stride stores: 3
         remark #15450: unmasked unaligned unit stride loads: 3
         remark #15455: masked aligned unit stride stores: 1
         remark #15456: masked unaligned unit stride loads: 1
         remark #15475: --- begin vector loop cost summary ---
         remark #15476: scalar loop cost: 78
         remark #15477: vector loop cost: 8.250
         remark #15478: estimated potential speedup: 7.000
         remark #15488: --- end vector loop cost summary ---
      LOOP END

      LOOP BEGIN at stencil.c(62,5) inlined into stencil.c(46,5)
      <Remainder loop for vectorization>
      LOOP END

      LOOP BEGIN at stencil.c(62,5) inlined into stencil.c(46,5)
      <Peeled loop for vectorization>
      LOOP END

      LOOP BEGIN at stencil.c(62,5) inlined into stencil.c(46,5)
         remark #15388: vectorization support: reference tmp_image has aligned
access   [ stencil.c(63,7) ]
         remark #15389: vectorization support: reference image has unaligned
access   [ stencil.c(63,7) ]
         remark #15388: vectorization support: reference tmp_image has aligned
access   [ stencil.c(65,21) ]
         remark #15388: vectorization support: reference tmp_image has aligned
access   [ stencil.c(65,21) ]
         remark #15389: vectorization support: reference image has unaligned
access   [ stencil.c(65,21) ]
         remark #15388: vectorization support: reference tmp_image has aligned
access   [ stencil.c(66,21) ]
         remark #15388: vectorization support: reference tmp_image has aligned
access   [ stencil.c(66,21) ]
         remark #15389: vectorization support: reference image has unaligned
access   [ stencil.c(66,21) ]
         remark #15388: vectorization support: reference tmp_image has aligned
access   [ stencil.c(67,21) ]
         remark #15388: vectorization support: reference tmp_image has aligned
access   [ stencil.c(67,21) ]
         remark #15389: vectorization support: reference image has unaligned
access   [ stencil.c(67,21) ]
         remark #15381: vectorization support: unaligned access used inside loop
body
         remark #15305: vectorization support: vector length 8
         remark #15309: vectorization support: normalized vectorization overhead
0.439
         remark #15300: LOOP WAS VECTORIZED
         remark #15442: entire loop may be executed in remainder
         remark #15448: unmasked aligned unit stride loads: 1
         remark #15449: unmasked aligned unit stride stores: 3
         remark #15450: unmasked unaligned unit stride loads: 3
         remark #15455: masked aligned unit stride stores: 1
         remark #15456: masked unaligned unit stride loads: 1
         remark #15475: --- begin vector loop cost summary ---
         remark #15476: scalar loop cost: 78
         remark #15477: vector loop cost: 8.250
         remark #15478: estimated potential speedup: 7.000
         remark #15488: --- end vector loop cost summary ---
      LOOP END

      LOOP BEGIN at stencil.c(62,5) inlined into stencil.c(46,5)
      <Remainder loop for vectorization>
      LOOP END

      LOOP BEGIN at stencil.c(62,5) inlined into stencil.c(46,5)
      <Peeled loop for vectorization>
      LOOP END

      LOOP BEGIN at stencil.c(62,5) inlined into stencil.c(46,5)
```

```
         remark #15388: vectorization support: reference tmp_image has aligned
access   [ stencil.c(63,7) ]
         remark #15389: vectorization support: reference image has unaligned
access   [ stencil.c(63,7) ]
         remark #15388: vectorization support: reference tmp_image has aligned
access   [ stencil.c(66,21) ]
         remark #15388: vectorization support: reference tmp_image has aligned
access   [ stencil.c(66,21) ]
         remark #15389: vectorization support: reference image has unaligned
access   [ stencil.c(66,21) ]
         remark #15388: vectorization support: reference tmp_image has aligned
access   [ stencil.c(67,21) ]
         remark #15388: vectorization support: reference tmp_image has aligned
access   [ stencil.c(67,21) ]
         remark #15389: vectorization support: reference image has unaligned
access   [ stencil.c(67,21) ]
         remark #15381: vectorization support: unaligned access used inside loop
body
         remark #15305: vectorization support: vector length 4
         remark #15309: vectorization support: normalized vectorization overhead
0.719
         remark #15300: LOOP WAS VECTORIZED
         remark #15442: entire loop may be executed in remainder
         remark #15448: unmasked aligned unit stride loads: 1
         remark #15449: unmasked aligned unit stride stores: 2
         remark #15450: unmasked unaligned unit stride loads: 2
         remark #15455: masked aligned unit stride stores: 1
         remark #15456: masked unaligned unit stride loads: 1
         remark #15475: --- begin vector loop cost summary ---
         remark #15476: scalar loop cost: 58
         remark #15477: vector loop cost: 8.000
         remark #15478: estimated potential speedup: 6.630
         remark #15488: --- end vector loop cost summary ---
      LOOP END

      LOOP BEGIN at stencil.c(62,5) inlined into stencil.c(46,5)
      <Remainder loop for vectorization>
      LOOP END
   LOOP END
LOOP END

LOOP BEGIN at stencil.c(113,5) inlined into stencil.c(56,3)
   remark #15542: loop was not vectorized: inner loop was already vectorized

   LOOP BEGIN at stencil.c(112,3) inlined into stencil.c(56,3)
   <Peeled loop for vectorization>
   LOOP END

   LOOP BEGIN at stencil.c(112,3) inlined into stencil.c(56,3)
      remark #15388: vectorization support: reference image has aligned access
[ stencil.c(114,7) ]
      remark #15305: vectorization support: vector length 8
      remark #15399: vectorization support: unroll factor set to 2
      remark #15309: vectorization support: normalized vectorization overhead 6.500
      remark #15301: PERMUTED LOOP WAS VECTORIZED
      remark #15442: entire loop may be executed in remainder
      remark #15448: unmasked aligned unit stride loads: 1
      remark #15475: --- begin vector loop cost summary ---
      remark #15476: scalar loop cost: 11
      remark #15477: vector loop cost: 0.620
      remark #15478: estimated potential speedup: 8.750
      remark #15488: --- end vector loop cost summary ---
   LOOP END

   LOOP BEGIN at stencil.c(112,3) inlined into stencil.c(56,3)
   <Remainder loop for vectorization>
```

```
      remark #15388: vectorization support: reference image has aligned access
[ stencil.c(114,7) ]
      remark #15305: vectorization support: vector length 8
      remark #15309: vectorization support: normalized vectorization overhead
13.000
      remark #15301: REMAINDER LOOP WAS VECTORIZED
   LOOP END

   LOOP BEGIN at stencil.c(112,3) inlined into stencil.c(56,3)
   <Remainder loop for vectorization>
   LOOP END
LOOP END

LOOP BEGIN at stencil.c(120,3) inlined into stencil.c(56,3)
   remark #15382: vectorization support: call to function fputc(int, FILE *)
cannot be vectorized   [ stencil.c(122,7) ]
   remark #15344: loop was not vectorized: vector dependence prevents vectorization
   remark #15346: vector dependence: assumed OUTPUT dependence between call:fputc
(int, FILE *) line 122 and call:fputc(int, FILE *) line 122
   remark #15346: vector dependence: assumed OUTPUT dependence between call:fputc
(int, FILE *) line 122 and call:fputc(int, FILE *) line 122

   LOOP BEGIN at stencil.c(121,5) inlined into stencil.c(56,3)
      remark #15527: loop was not vectorized: function call to fputc(int, FILE *)
cannot be vectorized   [ stencil.c(122,7) ]
   LOOP END
LOOP END
===========================================================================
```