```
TranslationUnitDecl 0x2efb270 <<invalid sloc>> <invalid sloc>
|-FunctionDecl 0x3055d10 <./CodeUnderTest.cpp:7:1, line:10:1> line:7:6 used random_fill 'void (float *, size_t)'
| | ParmVarDecl 0x3055b90 <col:18, col:25> col:25 used starting_vec 'float *'
| | ParmVarDecl 0x3055c00 <col:39, col:46> col:46 used size 'size_t':'unsigned long'
| | CompoundStmt 0x3056208 <col:51, line:10:1>
| |   `-ForStmt 0x30561a0 <line:8:5, line:9:40>
| |     |-DeclStmt 0x3055e60 <line:8:10, col:22>
| |     | `-VarDecl 0x3055dd0 <col:10, col:21> col:17 used i 'size_t':'unsigned long' cinit
| |     |   `-ImplicitCastExpr 0x3055e48 <col:21> 'size_t':'unsigned long' <IntegralCast>
| |     |     `-IntegerLiteral 0x3055e28 <col:21> 'int' 0
| |     |-<<<NULL>>>
| |     |-BinaryOperator 0x3055ef8 <col:24, col:28> '_Bool' '<'
| |     | |-ImplicitCastExpr 0x3055ec8 <col:24> 'size_t':'unsigned long' <LValueToRValue>
| |     | | `-DeclRefExpr 0x3055e78 <col:24> 'size_t':'unsigned long' lvalue Var 0x3055dd0 'i' 'size_t':'unsigned long'
| |     | `-ImplicitCastExpr 0x3055ee0 <col:28> 'size_t':'unsigned long' <LValueToRValue>
| |     |   `-DeclRefExpr 0x3055ea0 <col:28> 'size_t':'unsigned long' lvalue ParmVar 0x3055c00 'size' 'size_t':'unsigned long'
| |     |-UnaryOperator 0x3055f48 <col:34, col:36> 'size_t':'unsigned long' lvalue prefix '++'
| |     | `-DeclRefExpr 0x3055f20 <col:36> 'size_t':'unsigned long' lvalue Var 0x3055dd0 'i' 'size_t':'unsigned long'
| |     `-BinaryOperator 0x3056178 <line:9:9, col:40> 'float' lvalue '='
| |       |-ArraySubscriptExpr 0x3055fe8 <col:9, col:23> 'float' lvalue
| |       | |-ImplicitCastExpr 0x3055fb8 <col:9> 'float *' <LValueToRValue>
| |       | | `-DeclRefExpr 0x3055f68 <col:9> 'float *' lvalue ParmVar 0x3055b90 'starting_vec' 'float *'
| |       | `-ImplicitCastExpr 0x3055fd0 <col:22> 'size_t':'unsigned long' <LValueToRValue>
| |       |   `-DeclRefExpr 0x3055f90 <col:22> 'size_t':'unsigned long' lvalue Var 0x3055dd0 'i' 'size_t':'unsigned long'
| |       `-CStyleCastExpr 0x3056150 <col:27, col:40> 'float' <NoOp>
| |         `-ImplicitCastExpr 0x3056138 <col:35, col:40> 'float' <IntegralToFloating>
| |           `-CallExpr 0x3056100 <col:35, col:40> 'int'
| |             `-ImplicitCastExpr 0x30560e8 <col:35> 'int (*)(void) throw()' <FunctionToPointerDecay>
| |               `-DeclRefExpr 0x3056068 <col:35> 'int (void) throw()' lvalue Function 0x2fdfcc0 'rand' 'int (void) throw()'
| `-FullComment 0x305e640 <line:6:4, col:14>
|   `-ParagraphComment 0x305e610 <col:4, col:14>
|     `-TextComment 0x305e5e0 <col:4, col:14> Text="CAPA:IGNORE"
|-FunctionDecl 0x3059950 <line:16:1, line:20:1> line:16:6 used reshape2mat 'void (float *, float **, size_t)'
| | ParmVarDecl 0x3056240 <col:18, col:25> col:25 used in_vec 'float *'
| | ParmVarDecl 0x3056350 <col:33, col:48> col:40 used out_vec 'float **':'float **'
| | ParmVarDecl 0x30563c0 <col:51, col:58> col:58 used dim 'size_t':'unsigned long'
| | CompoundStmt 0x305a0c8 <col:62, line:20:1>
| |   `-ForStmt 0x305a088 <line:17:5, line:19:45>
| |     |-DeclStmt 0x3059ab0 <line:17:10, col:22>
| |     | `-VarDecl 0x3059a20 <col:10, col:21> col:17 used i 'size_t':'unsigned long' cinit
| |     |   `-ImplicitCastExpr 0x3059a98 <col:21> 'size_t':'unsigned long' <IntegralCast>
| |     |     `-IntegerLiteral 0x3059a78 <col:21> 'int' 0
| |     |-<<<NULL>>>
| |     |-BinaryOperator 0x3059b48 <col:24, col:28> '_Bool' '<'
| |     | |-ImplicitCastExpr 0x3059b18 <col:24> 'size_t':'unsigned long' <LValueToRValue>
| |     | | `-DeclRefExpr 0x3059ac8 <col:24> 'size_t':'unsigned long' lvalue Var 0x3059a20 'i' 'size_t':'unsigned long'
| |     | `-ImplicitCastExpr 0x3059b30 <col:28> 'size_t':'unsigned long' <LValueToRValue>
| |     |   `-DeclRefExpr 0x3059af0 <col:28> 'size_t':'unsigned long' lvalue ParmVar 0x30563c0 'dim' 'size_t':'unsigned long'
| |     |-UnaryOperator 0x3059b98 <col:33, col:35> 'size_t':'unsigned long' lvalue prefix '++'
| |     | `-DeclRefExpr 0x3059b70 <col:35> 'size_t':'unsigned long' lvalue Var 0x3059a20 'i' 'size_t':'unsigned long'
| |     `-ForStmt 0x305a048 <line:18:9, line:19:45>
| |       |-DeclStmt 0x3059c60 <line:18:14, col:26>
| |       | `-VarDecl 0x3059bd0 <col:14, col:25> col:21 used j 'size_t':'unsigned long' cinit
| |       |   `-ImplicitCastExpr 0x3059c48 <col:25> 'size_t':'unsigned long' <IntegralCast>
| |       |     `-IntegerLiteral 0x3059c28 <col:25> 'int' 0
| |       |-<<<NULL>>>
| |       |-BinaryOperator 0x3059cf8 <col:28, col:32> '_Bool' '<'
| |       | |-ImplicitCastExpr 0x3059cc8 <col:28> 'size_t':'unsigned long' <LValueToRValue>
| |       | | `-DeclRefExpr 0x3059c78 <col:28> 'size_t':'unsigned long' lvalue Var 0x3059bd0 'j' 'size_t':'unsigned long'
| |       | `-ImplicitCastExpr 0x3059ce0 <col:32> 'size_t':'unsigned long' <LValueToRValue>
| |       |   `-DeclRefExpr 0x3059ca0 <col:32> 'size_t':'unsigned long' lvalue ParmVar 0x30563c0 'dim' 'size_t':'unsigned long'
| |       |-UnaryOperator 0x3059d48 <col:37, col:39> 'size_t':'unsigned long' lvalue prefix '++'
| |       | `-DeclRefExpr 0x3059d20 <col:39> 'size_t':'unsigned long' lvalue Var 0x3059bd0 'j' 'size_t':'unsigned long'
| |       `-BinaryOperator 0x305a020 <line:19:13, col:45> 'float' lvalue '='
| |         |-ArraySubscriptExpr 0x3059e68 <col:13, col:25> 'float' lvalue
| |         | |-ImplicitCastExpr 0x3059e38 <col:13, col:22> 'float *' <LValueToRValue>
| |         | | `-ArraySubscriptExpr 0x3059de8 <col:13, col:22> 'float *' lvalue
| |         | |   |-ImplicitCastExpr 0x3059db8 <col:13> 'float **':'float **' <LValueToRValue>
| |         | |   | `-DeclRefExpr 0x3059d68 <col:13> 'float **':'float **' lvalue ParmVar 0x3056350 'out_vec' 'float **':'float **'
| |         | |   `-ImplicitCastExpr 0x3059dd0 <col:21> 'size_t':'unsigned long' <LValueToRValue>
| |         | |     `-DeclRefExpr 0x3059d90 <col:21> 'size_t':'unsigned long' lvalue Var 0x3059a20 'i' 'size_t':'unsigned long'
| |         | `-ImplicitCastExpr 0x3059e50 <col:24> 'size_t':'unsigned long' <LValueToRValue>
| |         |   `-DeclRefExpr 0x3059e10 <col:24> 'size_t':'unsigned long' lvalue Var 0x3059bd0 'j' 'size_t':'unsigned long'
| |         `-ImplicitCastExpr 0x305a008 <col:29, col:45> 'float' <LValueToRValue>
| |           `-ArraySubscriptExpr 0x3059fe0 <col:29, col:45> 'float' lvalue
| |             |-ImplicitCastExpr 0x3059fc8 <col:29> 'float *' <LValueToRValue>
| |             | `-DeclRefExpr 0x3059e90 <col:29> 'float *' lvalue ParmVar 0x3056240 'in_vec' 'float *'
| |             `-BinaryOperator 0x3059fa0 <col:36, col:44> 'unsigned long' '+'
| |               |-BinaryOperator 0x3059f38 <col:36, col:38> 'unsigned long' '*'
| |               | |-ImplicitCastExpr 0x3059f08 <col:36> 'size_t':'unsigned long' <LValueToRValue>
| |               | | `-DeclRefExpr 0x3059eb8 <col:36> 'size_t':'unsigned long' lvalue Var 0x3059a20 'i' 'size_t':'unsigned long'
| |               | `-ImplicitCastExpr 0x3059f20 <col:38> 'size_t':'unsigned long' <LValueToRValue>
| |               |   `-DeclRefExpr 0x3059ee0 <col:38> 'size_t':'unsigned long' lvalue ParmVar 0x30563c0 'dim' 'size_t':'unsigned long'
| |               `-ImplicitCastExpr 0x3059f88 <col:44> 'size_t':'unsigned long' <LValueToRValue>
| |                 `-DeclRefExpr 0x3059f60 <col:44> 'size_t':'unsigned long' lvalue Var 0x3059bd0 'j' 'size_t':'unsigned long'
| `-FullComment 0x305e760 <line:12:4, line:14:14>
|   `-ParagraphComment 0x305e730 <line:12:4, line:14:14>
|     |-TextComment 0x305e6b0 <line:12:4, col:54> Text=" This Function is reponsible for reshaping a vector"
|     |-TextComment 0x305e6d0 <line:13:3, col:17> Text=" into a matrix."
|     `-TextComment 0x305e6f0 <line:14:3, col:14> Text=" CAPA:IGNORE"
|-FunctionDecl 0x305a280 <line:22:1, line:26:1> line:22:6 used reshape2vec 'void (float *, float **, size_t)'
| | ParmVarDecl 0x305a100 <col:18, col:25> col:25 used out_vec 'float *'
| | ParmVarDecl 0x305a180 <col:34, col:48> col:41 used in_mat 'float **':'float **'
| | ParmVarDecl 0x305a1f0 <col:51, col:58> col:58 used dim 'size_t':'unsigned long'
| | CompoundStmt 0x305aa38 <col:62, line:26:1>
| |   `-ForStmt 0x305a9c8 <line:23:5, line:25:45>
| |     |-DeclStmt 0x305a3e0 <line:23:10, col:22>
| |     | `-VarDecl 0x305a350 <col:10, col:21> col:17 used i 'size_t':'unsigned long' cinit
| |     |   `-ImplicitCastExpr 0x305a3c8 <col:21> 'size_t':'unsigned long' <IntegralCast>
| |     |     `-IntegerLiteral 0x305a3a8 <col:21> 'int' 0
| |     |-<<<NULL>>>
| |     |-BinaryOperator 0x305a478 <col:24, col:28> '_Bool' '<'
| |     | |-ImplicitCastExpr 0x305a448 <col:24> 'size_t':'unsigned long' <LValueToRValue>
| |     | | `-DeclRefExpr 0x305a3f8 <col:24> 'size_t':'unsigned long' lvalue Var 0x305a350 'i' 'size_t':'unsigned long'
| |     | `-ImplicitCastExpr 0x305a460 <col:28> 'size_t':'unsigned long' <LValueToRValue>
| |     |   `-DeclRefExpr 0x305a420 <col:28> 'size_t':'unsigned long' lvalue ParmVar 0x305a1f0 'dim' 'size_t':'unsigned long'
| |     |-UnaryOperator 0x305a4c8 <col:33, col:35> 'size_t':'unsigned long' lvalue prefix '++'
```

```
|   |   `-DeclRefExpr 0x305a4a0 <col:35> 'size_t':'unsigned long' lvalue Var 0x305a350 'i' 'size_t':'unsigned long'
|   `-ForStmt 0x305a988 <line:24:9, line:25:45>
|     |-DeclStmt 0x305a590 <line:24:14, col:26>
|     | `-VarDecl 0x305a500 <col:14, col:25> col:21 used j 'size_t':'unsigned long' cinit
|     |   `-ImplicitCastExpr 0x305a578 <col:25> 'size_t':'unsigned long' <IntegralCast>
|     |     `-IntegerLiteral 0x305a558 <col:25> 'int' 0
|     |-<<<NULL>>>
|     |-BinaryOperator 0x305a628 <col:28, col:32> '_Bool' '<'
|     | |-ImplicitCastExpr 0x305a5f8 <col:28> 'size_t':'unsigned long' <LValueToRValue>
|     | | `-DeclRefExpr 0x305a5a8 <col:28> 'size_t':'unsigned long' lvalue Var 0x305a500 'j' 'size_t':'unsigned long'
|     | `-ImplicitCastExpr 0x305a610 <col:32> 'size_t':'unsigned long' <LValueToRValue>
|     |   `-DeclRefExpr 0x305a5d0 <col:32> 'size_t':'unsigned long' lvalue ParmVar 0x305a1f0 'dim' 'size_t':'unsigned long'
|     |-UnaryOperator 0x305a678 <col:37, col:39> 'size_t':'unsigned long' lvalue prefix '++'
|     | `-DeclRefExpr 0x305a650 <col:39> 'size_t':'unsigned long' lvalue Var 0x305a500 'j' 'size_t':'unsigned long'
|     `-BinaryOperator 0x305a960 <line:25:13, col:45> 'float' lvalue '='
|       |-ArraySubscriptExpr 0x305a7e8 <col:13, col:30> 'float' lvalue
|       | |-ImplicitCastExpr 0x305a7d0 <col:13> 'float *' <LValueToRValue>
|       | | `-DeclRefExpr 0x305a698 <col:13> 'float *' lvalue ParmVar 0x305a100 'out_vec' 'float *'
|       | `-BinaryOperator 0x305a7a8 <col:21, col:29> 'unsigned long' '+'
|       |   |-BinaryOperator 0x305a740 <col:21, col:23> 'unsigned long' '*'
|       |   | |-ImplicitCastExpr 0x305a710 <col:21> 'size_t':'unsigned long' <LValueToRValue>
|       |   | | `-DeclRefExpr 0x305a6c0 <col:21> 'size_t':'unsigned long' lvalue Var 0x305a350 'i' 'size_t':'unsigned long'
|       |   | `-ImplicitCastExpr 0x305a728 <col:23> 'size_t':'unsigned long' <LValueToRValue>
|       |   |   `-DeclRefExpr 0x305a6e8 <col:23> 'size_t':'unsigned long' lvalue ParmVar 0x305a1f0 'dim' 'size_t':'unsigned long'
|       |   `-ImplicitCastExpr 0x305a790 <col:29> 'size_t':'unsigned long' <LValueToRValue>
|       |     `-DeclRefExpr 0x305a768 <col:29> 'size_t':'unsigned long' lvalue Var 0x305a500 'j' 'size_t':'unsigned long'
|       `-ImplicitCastExpr 0x305a948 <col:34, col:45> 'float' <LValueToRValue>
|         `-ArraySubscriptExpr 0x305a920 <col:34, col:45> 'float' lvalue
|           |-ImplicitCastExpr 0x305a8e0 <col:34, col:42> 'float *' <LValueToRValue>
|           | `-ArraySubscriptExpr 0x305a890 <col:34, col:42> 'float *' lvalue
|           |   |-ImplicitCastExpr 0x305a860 <col:34> 'float **':'float **' <LValueToRValue>
|           |   | `-DeclRefExpr 0x305a810 <col:34> 'float **':'float **' lvalue ParmVar 0x305a180 'in_mat' 'float **':'float **'
|           |   `-ImplicitCastExpr 0x305a878 <col:41> 'size_t':'unsigned long' <LValueToRValue>
|           |     `-DeclRefExpr 0x305a838 <col:41> 'size_t':'unsigned long' lvalue Var 0x305a350 'i' 'size_t':'unsigned long'
|           `-ImplicitCastExpr 0x305a8f8 <col:44> 'size_t':'unsigned long' <LValueToRValue>
|             `-DeclRefExpr 0x305a8b8 <col:44> 'size_t':'unsigned long' lvalue Var 0x305a500 'j' 'size_t':'unsigned long'
|-FunctionDecl 0x305ad00 <line:29:1, col:55> col:6 used mmult 'void (float **, float **, float **, size_t)'
| |-ParmVarDecl 0x305aa70 <col:12, col:20> col:20 A 'float **'
| |-ParmVarDecl 0x305aae0 <col:23, col:31> col:31 B 'float **'
| |-ParmVarDecl 0x305ab50 <col:34, col:42> col:42 C 'float **'
| |-ParmVarDecl 0x305abc0 <col:45, col:52> col:52 dim 'size_t':'unsigned long'
| `-FullComment 0x305e830 <line:28:4, col:15>
|   `-ParagraphComment 0x305e800 <col:4, col:15>
|     `-TextComment 0x305e7d0 <col:4, col:15> Text=" CAPA:IGNORE"
`-FunctionDecl 0x305ade0 <line:31:1, line:74:1> line:31:5 main 'int (void)'
  `-CompoundStmt 0x305d810 <line:32:1, line:74:1>
    |-DeclStmt 0x305afb8 <line:34:5, col:35>
    | `-VarDecl 0x305ae90 <col:5, col:31> col:18 referenced ELEMS 'const size_t':'const unsigned long' cinit
    |   `-ImplicitCastExpr 0x305af50 <col:26, col:31> 'const size_t':'const unsigned long' <IntegralCast>
    |     `-BinaryOperator 0x305af28 <col:26, col:31> 'int' '*'
    |       |-IntegerLiteral 0x305aee8 <col:26> 'int' 1000
    |       `-IntegerLiteral 0x305af08 <col:31> 'int' 1000
    |-DeclStmt 0x305b0c8 <line:35:5, col:30>
    | `-VarDecl 0x305b070 <col:5, col:29> col:11 used starting_vec 'float [1000000]'
    |-DeclStmt 0x305b148 <line:36:5, col:13>
    | `-VarDecl 0x305b0f0 <col:5, col:12> col:12 used t 'time_t':'long'
    |-CallExpr 0x305b410 <line:38:5, col:30> 'void'
    | |-ImplicitCastExpr 0x305b3f8 <col:5> 'void (*)(unsigned int) throw()' <FunctionToPointerDecay>
    | | `-DeclRefExpr 0x305b370 <col:5> 'void (unsigned int) throw()' lvalue Function 0x2fdfdf0 'srand' 'void (unsigned int) throw()'
    | `-CStyleCastExpr 0x305b348 <col:11, col:29> 'unsigned int' <NoOp>
    |   `-ImplicitCastExpr 0x305b330 <col:22, col:29> 'unsigned int' <IntegralCast>
    |     `-CallExpr 0x305b2f0 <col:22, col:29> 'time_t':'long'
    |       |-ImplicitCastExpr 0x305b2d8 <col:22> 'time_t (*)(time_t *) throw()' <FunctionToPointerDecay>
    |       | `-DeclRefExpr 0x305b258 <col:22> 'time_t (time_t *) throw()' lvalue Function 0x3001960 'time' 'time_t (time_t *) throw()'
    |       `-UnaryOperator 0x305b238 <col:27, col:28> 'time_t *' prefix '&'
    |         `-DeclRefExpr 0x305b210 <col:28> 'time_t':'long' lvalue Var 0x305b0f0 't' 'time_t':'long'
    |-CallExpr 0x305b580 <line:39:5, col:36> 'void'
    | |-ImplicitCastExpr 0x305b568 <col:5> 'void (*)(float *, size_t)' <FunctionToPointerDecay>
    | | `-DeclRefExpr 0x305b4e8 <col:5> 'void (float *, size_t)' lvalue Function 0x3055d10 'random_fill' 'void (float *, size_t)'
    | |-ImplicitCastExpr 0x305b5b8 <col:17> 'float *' <ArrayToPointerDecay>
    | | `-DeclRefExpr 0x305b498 <col:17> 'float [1000000]' lvalue Var 0x305b070 'starting_vec' 'float [1000000]'
    | `-ImplicitCastExpr 0x305b5d0 <col:31> 'size_t':'unsigned long' <LValueToRValue>
    |   `-DeclRefExpr 0x305b4c0 <col:31> 'const size_t':'const unsigned long' lvalue Var 0x305ae90 'ELEMS' 'const size_t':'const unsigned long'
    |-ForStmt 0x305ba08 <line:42:5, line:45:5>
    | |-DeclStmt 0x305b690 <line:42:10, col:22>
    | | `-VarDecl 0x305b600 <col:10, col:21> col:17 used i 'size_t':'unsigned long' cinit
    | |   `-ImplicitCastExpr 0x305b678 <col:21> 'size_t':'unsigned long' <IntegralCast>
    | |     `-IntegerLiteral 0x305b658 <col:21> 'int' 0
    | |-<<<NULL>>>
    | |-BinaryOperator 0x305b728 <col:24, col:28> '_Bool' '<'
    | | |-ImplicitCastExpr 0x305b6f8 <col:24> 'size_t':'unsigned long' <LValueToRValue>
    | | | `-DeclRefExpr 0x305b6a8 <col:24> 'size_t':'unsigned long' lvalue Var 0x305b600 'i' 'size_t':'unsigned long'
    | | `-ImplicitCastExpr 0x305b710 <col:28> 'size_t':'unsigned long' <LValueToRValue>
    | |   `-DeclRefExpr 0x305b6d0 <col:28> 'const size_t':'const unsigned long' lvalue Var 0x305ae90 'ELEMS' 'const size_t':'const unsigned long'
    | |-UnaryOperator 0x305b778 <col:35, col:37> 'size_t':'unsigned long' lvalue prefix '++'
    | | `-DeclRefExpr 0x305b750 <col:37> 'size_t':'unsigned long' lvalue Var 0x305b600 'i' 'size_t':'unsigned long'
    | `-CompoundStmt 0x305b9e0 <col:39, line:45:5>
    |   |-CompoundAssignOperator 0x305b878 <line:43:9, col:28> 'float' lvalue '/=' ComputeLHSTy='float' ComputeResultTy='float'
    |   | |-ArraySubscriptExpr 0x305b818 <col:9, col:23> 'float' lvalue
    |   | | |-ImplicitCastExpr 0x305b7e8 <col:9> 'float *' <ArrayToPointerDecay>
    |   | | | `-DeclRefExpr 0x305b798 <col:9> 'float [1000000]' lvalue Var 0x305b070 'starting_vec' 'float [1000000]'
    |   | | `-ImplicitCastExpr 0x305b800 <col:22> 'size_t':'unsigned long' <LValueToRValue>
    |   | |   `-DeclRefExpr 0x305b7c0 <col:22> 'size_t':'unsigned long' lvalue Var 0x305b600 'i' 'size_t':'unsigned long'
    |   | `-ImplicitCastExpr 0x305b860 <col:28> 'float' <IntegralToFloating>
    |   |   `-IntegerLiteral 0x305b840 <col:28> 'int' 2
    |   `-CompoundAssignOperator 0x305b9a8 <line:44:9, col:28> 'float' lvalue '+=' ComputeLHSTy='float' ComputeResultTy='float'
    |     |-ArraySubscriptExpr 0x305b948 <col:9, col:23> 'float' lvalue
    |     | |-ImplicitCastExpr 0x305b900 <col:9> 'float *' <ArrayToPointerDecay>
    |     | | `-DeclRefExpr 0x305b8b0 <col:9> 'float [1000000]' lvalue Var 0x305b070 'starting_vec' 'float [1000000]'
    |     | `-ImplicitCastExpr 0x305b930 <col:22> 'size_t':'unsigned long' <LValueToRValue>
    |     |   `-DeclRefExpr 0x305b8d8 <col:22> 'size_t':'unsigned long' lvalue Var 0x305b600 'i' 'size_t':'unsigned long'
    |     `-ImplicitCastExpr 0x305b990 <col:28> 'float' <IntegralToFloating>
    |       `-IntegerLiteral 0x305b970 <col:28> 'int' 4
    |-DeclStmt 0x305baf0 <line:48:5, col:16>
    | `-VarDecl 0x305ba60 <col:5, col:15> col:11 used k 'float' cinit
    |   `-ImplicitCastExpr 0x305bad8 <col:15> 'float' <IntegralToFloating>
    |     `-IntegerLiteral 0x305bab8 <col:15> 'int' 0
```

```
|-ForStmt 0x305be58 <line:49:5, line:50:30>
| |-DeclStmt 0x305bbb0 <line:49:10, col:22>
| | `-VarDecl 0x305bb20 <col:10, col:21> col:17 used i 'size_t':'unsigned long' cinit
| |   `-ImplicitCastExpr 0x305bb98 <col:21> 'size_t':'unsigned long' <IntegralCast>
| |     `-IntegerLiteral 0x305bb78 <col:21> 'int' 0
| |-<<<NULL>>>
| |-BinaryOperator 0x305bc48 <col:24, col:28> '_Bool' '<'
| | |-ImplicitCastExpr 0x305bc18 <col:24> 'size_t':'unsigned long' <LValueToRValue>
| | | `-DeclRefExpr 0x305bbc8 <col:24> 'size_t':'unsigned long' lvalue Var 0x305bb20 'i' 'size_t':'unsigned long'
| | `-ImplicitCastExpr 0x305bc30 <col:28> 'size_t':'unsigned long' <LValueToRValue>
| |   `-DeclRefExpr 0x305bbf0 <col:28> 'const size_t':'const unsigned long' lvalue Var 0x305ae90 'ELEMS' 'const size_t':'const unsigned long'
| |-UnaryOperator 0x305bc98 <col:35, col:37> 'size_t':'unsigned long' lvalue prefix '++'
| | `-DeclRefExpr 0x305bc70 <col:37> 'size_t':'unsigned long' lvalue Var 0x305bb20 'i' 'size_t':'unsigned long'
| `-CompoundAssignOperator 0x305be20 <line:50:9, col:30> 'float' lvalue '+=' ComputeLHSTy='float' ComputeResultTy='float'
|   |-DeclRefExpr 0x305bcb8 <col:9> 'float' lvalue Var 0x305ba60 'k' 'float'
|   `-BinaryOperator 0x305bdf8 <col:14, col:30> 'float' '/'
|     |-ImplicitCastExpr 0x305bdb0 <col:14, col:28> 'float' <LValueToRValue>
|     | `-ArraySubscriptExpr 0x305bd60 <col:14, col:28> 'float' lvalue
|     |   |-ImplicitCastExpr 0x305bd30 <col:14> 'float *' <ArrayToPointerDecay>
|     |   | `-DeclRefExpr 0x305bce0 <col:14> 'float [1000000]' lvalue Var 0x305b070 'starting_vec' 'float [1000000]'
|     |   `-ImplicitCastExpr 0x305bd48 <col:27> 'size_t':'unsigned long' <LValueToRValue>
|     |     `-DeclRefExpr 0x305bd08 <col:27> 'size_t':'unsigned long' lvalue Var 0x305bb20 'i' 'size_t':'unsigned long'
|     `-ImplicitCastExpr 0x305bde0 <col:30> 'float' <IntegralToFloating>
|       `-ImplicitCastExpr 0x305bdc8 <col:30> 'size_t':'unsigned long' <LValueToRValue>
|         `-DeclRefExpr 0x305bd88 <col:30> 'const size_t':'const unsigned long' lvalue Var 0x305ae90 'ELEMS' 'const size_t':'const unsigned long'
|-DeclStmt 0x305bf58 <line:53:5, col:25>
| `-VarDecl 0x305bf00 <col:5, col:24> col:11 used cum_sum 'float [1000000]'
|-BinaryOperator 0x305c100 <line:54:5, col:34> 'float' lvalue '='
| |-ArraySubscriptExpr 0x305bfd0 <col:5, col:14> 'float' lvalue
| | |-ImplicitCastExpr 0x305bfb8 <col:5> 'float *' <ArrayToPointerDecay>
| | | `-DeclRefExpr 0x305bf70 <col:5> 'float [1000000]' lvalue Var 0x305bf00 'cum_sum' 'float [1000000]'
| | `-IntegerLiteral 0x305bf98 <col:13> 'int' 0
| `-BinaryOperator 0x305c0d8 <col:18, col:34> 'float' '/'
|   |-ImplicitCastExpr 0x305c0a8 <col:18, col:32> 'float' <LValueToRValue>
|   | `-ArraySubscriptExpr 0x305c058 <col:18, col:32> 'float' lvalue
|   |   |-ImplicitCastExpr 0x305c040 <col:18> 'float *' <ArrayToPointerDecay>
|   |   | `-DeclRefExpr 0x305bff8 <col:18> 'float [1000000]' lvalue Var 0x305b070 'starting_vec' 'float [1000000]'
|   |   `-IntegerLiteral 0x305c020 <col:31> 'int' 0
|   `-ImplicitCastExpr 0x305c0c0 <col:34> 'float' <LValueToRValue>
|     `-DeclRefExpr 0x305c080 <col:34> 'float' lvalue Var 0x305ba60 'k' 'float'
|-ForStmt 0x305c630 <line:55:5, line:56:57>
| |-DeclStmt 0x305c1d0 <line:55:10, col:22>
| | `-VarDecl 0x305c140 <col:10, col:21> col:17 used i 'size_t':'unsigned long' cinit
| |   `-ImplicitCastExpr 0x305c1b8 <col:21> 'size_t':'unsigned long' <IntegralCast>
| |     `-IntegerLiteral 0x305c198 <col:21> 'int' 1
| |-<<<NULL>>>
| |-BinaryOperator 0x305c268 <col:24, col:28> '_Bool' '<'
| | |-ImplicitCastExpr 0x305c238 <col:24> 'size_t':'unsigned long' <LValueToRValue>
| | | `-DeclRefExpr 0x305c1e8 <col:24> 'size_t':'unsigned long' lvalue Var 0x305c140 'i' 'size_t':'unsigned long'
| | `-ImplicitCastExpr 0x305c250 <col:28> 'size_t':'unsigned long' <LValueToRValue>
| |   `-DeclRefExpr 0x305c210 <col:28> 'const size_t':'const unsigned long' lvalue Var 0x305ae90 'ELEMS' 'const size_t':'const unsigned long'
| |-UnaryOperator 0x305c2b8 <col:35, col:36> 'size_t':'unsigned long' postfix '++'
| | `-DeclRefExpr 0x305c290 <col:35> 'size_t':'unsigned long' lvalue Var 0x305c140 'i' 'size_t':'unsigned long'
| `-BinaryOperator 0x305c608 <line:56:9, col:57> 'float' lvalue '='
|   |-ArraySubscriptExpr 0x305c358 <col:9, col:18> 'float' lvalue
|   | |-ImplicitCastExpr 0x305c328 <col:9> 'float *' <ArrayToPointerDecay>
|   | | `-DeclRefExpr 0x305c2d8 <col:9> 'float [1000000]' lvalue Var 0x305bf00 'cum_sum' 'float [1000000]'
|   | `-ImplicitCastExpr 0x305c340 <col:17> 'size_t':'unsigned long' <LValueToRValue>
|   |   `-DeclRefExpr 0x305c300 <col:17> 'size_t':'unsigned long' lvalue Var 0x305c140 'i' 'size_t':'unsigned long'
|   `-BinaryOperator 0x305c5e0 <col:22, col:57> 'float' '+'
|     |-BinaryOperator 0x305c498 <col:22, col:38> 'float' '/'
|     | |-ImplicitCastExpr 0x305c450 <col:22, col:36> 'float' <LValueToRValue>
|     | | `-ArraySubscriptExpr 0x305c400 <col:22, col:36> 'float' lvalue
|     | |   |-ImplicitCastExpr 0x305c3d0 <col:22> 'float *' <ArrayToPointerDecay>
|     | |   | `-DeclRefExpr 0x305c380 <col:22> 'float [1000000]' lvalue Var 0x305b070 'starting_vec' 'float [1000000]'
|     | |   `-ImplicitCastExpr 0x305c3e8 <col:35> 'size_t':'unsigned long' <LValueToRValue>
|     | |     `-DeclRefExpr 0x305c3a8 <col:35> 'size_t':'unsigned long' lvalue Var 0x305c140 'i' 'size_t':'unsigned long'
|     | `-ImplicitCastExpr 0x305c480 <col:38> 'float' <IntegralToFloating>
|     |   `-ImplicitCastExpr 0x305c468 <col:38> 'size_t':'unsigned long' <LValueToRValue>
|     |     `-DeclRefExpr 0x305c428 <col:38> 'const size_t':'const unsigned long' lvalue Var 0x305ae90 'ELEMS' 'const size_t':'const unsigned long'
|     `-ImplicitCastExpr 0x305c5c8 <col:46, col:57> 'float' <LValueToRValue>
|       `-ArraySubscriptExpr 0x305c5a0 <col:46, col:57> 'float' lvalue
|         |-ImplicitCastExpr 0x305c588 <col:46> 'float *' <ArrayToPointerDecay>
|         | `-DeclRefExpr 0x305c4c0 <col:46> 'float [1000000]' lvalue Var 0x305bf00 'cum_sum' 'float [1000000]'
|         `-BinaryOperator 0x305c560 <col:54, col:56> 'unsigned long' '-'
|           |-ImplicitCastExpr 0x305c530 <col:54> 'size_t':'unsigned long' <LValueToRValue>
|           | `-DeclRefExpr 0x305c4e8 <col:54> 'size_t':'unsigned long' lvalue Var 0x305c140 'i' 'size_t':'unsigned long'
|           `-ImplicitCastExpr 0x305c548 <col:56> 'unsigned long' <IntegralCast>
|             `-IntegerLiteral 0x305c510 <col:56> 'int' 1
|-DeclStmt 0x305c8b8 <line:59:5, col:35>
| `-VarDecl 0x305c680 <col:5, col:34> col:18 used dim 'const size_t':'const unsigned long' cinit
|   `-ImplicitCastExpr 0x305c850 <col:24, col:34> 'const size_t':'const unsigned long' <FloatingToIntegral>
|     `-CallExpr 0x305c7f0 <col:24, col:34> 'double'
|       |-ImplicitCastExpr 0x305c7d8 <col:24> 'double (*)(double) throw()' <FunctionToPointerDecay>
|       | `-DeclRefExpr 0x305c758 <col:24> 'double (double) throw()' lvalue Function 0x2fc3370 'sqrt' 'double (double) throw()'
|       `-ImplicitCastExpr 0x305c838 <col:29> 'double' <IntegralToFloating>
|         `-ImplicitCastExpr 0x305c820 <col:29> 'size_t':'unsigned long' <LValueToRValue>
|           `-DeclRefExpr 0x305c730 <col:29> 'const size_t':'const unsigned long' lvalue Var 0x305ae90 'ELEMS' 'const size_t':'const unsigned long'
|-DeclStmt 0x305ca78 <line:60:5, col:32>
| `-VarDecl 0x305ca20 <col:5, col:31> col:11 used cum_sum_mat 'float [dim][dim]'
|-CallExpr 0x305cc80 <line:62:5, col:53> 'void'
| |-ImplicitCastExpr 0x305cc68 <col:5> 'void (*)(float *, float **, size_t)' <FunctionToPointerDecay>
| | `-DeclRefExpr 0x305cbe0 <col:5> 'void (float *, float **, size_t)' lvalue Function 0x3059950 'reshape2mat' 'void (float *, float **, size_t)'
| |-ImplicitCastExpr 0x305ccc0 <col:17> 'float *' <ArrayToPointerDecay>
| | `-DeclRefExpr 0x305cae8 <col:17> 'float [1000000]' lvalue Var 0x305bf00 'cum_sum' 'float [1000000]'
| |-CStyleCastExpr 0x305cb90 <col:26, col:37> 'float **' <BitCast>
| | `-ImplicitCastExpr 0x305cb78 <col:37> 'float (*)[dim]' <ArrayToPointerDecay>
| |   `-DeclRefExpr 0x305cb10 <col:37> 'float [dim][dim]' lvalue Var 0x305ca20 'cum_sum_mat' 'float [dim][dim]'
| `-ImplicitCastExpr 0x305ccd8 <col:50> 'size_t':'unsigned long' <LValueToRValue>
|   `-DeclRefExpr 0x305cbb8 <col:50> 'const size_t':'const unsigned long' lvalue Var 0x305c680 'dim' 'const size_t':'const unsigned long'
|-CallExpr 0x305cf90 <line:63:5, col:86> 'void'
| |-ImplicitCastExpr 0x305cf78 <col:5> 'void (*)(float **, float **, float **, size_t)' <FunctionToPointerDecay>
| | `-DeclRefExpr 0x305cef0 <col:5> 'void (float **, float **, float **, size_t)' lvalue Function 0x305ad00 'mmult' 'void (float **, float **, float **, size_t)'
| |-CStyleCastExpr 0x305cda0 <col:11, col:22> 'float **' <BitCast>
| | `-ImplicitCastExpr 0x305cd88 <col:22> 'float (*)[dim]' <ArrayToPointerDecay>
| |   `-DeclRefExpr 0x305cd48 <col:22> 'float [dim][dim]' lvalue Var 0x305ca20 'cum_sum_mat' 'float [dim][dim]'
| |-CStyleCastExpr 0x305ce20 <col:35, col:46> 'float **' <BitCast>
```

```
|   | | `-ImplicitCastExpr 0x305ce08 <col:46> 'float (*)[dim]' <ArrayToPointerDecay>
|   | |   `-DeclRefExpr 0x305cdc8 <col:46> 'float [dim][dim]' lvalue Var 0x305ca20 'cum_sum_mat' 'float [dim][dim]'
|   | |-CStyleCastExpr 0x305cea0 <col:59, col:70> 'float **' <BitCast>
|   | | `-ImplicitCastExpr 0x305ce88 <col:70> 'float (*)[dim]' <ArrayToPointerDecay>
|   | |   `-DeclRefExpr 0x305ce48 <col:70> 'float [dim][dim]' lvalue Var 0x305ca20 'cum_sum_mat' 'float [dim][dim]'
|   | `-ImplicitCastExpr 0x305cfd8 <col:83> 'size_t':'unsigned long' <LValueToRValue>
|   |   `-DeclRefExpr 0x305cec8 <col:83> 'const size_t':'const unsigned long' lvalue Var 0x305c680 'dim' 'const size_t':'const unsigned long'
|   |-CallExpr 0x305d158 <line:64:5, col:53> 'void'
|   | |-ImplicitCastExpr 0x305d140 <col:5> 'void (*)(float *, float **, size_t)' <FunctionToPointerDecay>
|   | | `-DeclRefExpr 0x305d118 <col:5> 'void (float *, float **, size_t)' lvalue Function 0x305a280 'reshape2vec' 'void (float *, float **, size_t)'
|   | |-ImplicitCastExpr 0x305d198 <col:17> 'float *' <ArrayToPointerDecay>
|   | | `-DeclRefExpr 0x305d048 <col:17> 'float [1000000]' lvalue Var 0x305bf00 'cum_sum' 'float [1000000]'
|   | |-CStyleCastExpr 0x305d0c8 <col:26, col:37> 'float **' <BitCast>
|   | | `-ImplicitCastExpr 0x305d0b0 <col:37> 'float (*)[dim]' <ArrayToPointerDecay>
|   | |   `-DeclRefExpr 0x305d070 <col:37> 'float [dim][dim]' lvalue Var 0x305ca20 'cum_sum_mat' 'float [dim][dim]'
|   | `-ImplicitCastExpr 0x305d1b0 <col:50> 'size_t':'unsigned long' <LValueToRValue>
|   |   `-DeclRefExpr 0x305d0f0 <col:50> 'const size_t':'const unsigned long' lvalue Var 0x305c680 'dim' 'const size_t':'const unsigned long'
|   |-BinaryOperator 0x305d228 <line:68:5, col:9> 'float' lvalue '='
|   | |-DeclRefExpr 0x305d1c8 <col:5> 'float' lvalue Var 0x305ba60 'k' 'float'
|   | `-ImplicitCastExpr 0x305d210 <col:9> 'float' <IntegralToFloating>
|   |   `-IntegerLiteral 0x305d1f0 <col:9> 'int' 0
|   |-ForStmt 0x305d758 <line:69:5, line:71:38>
|   | |-DeclStmt 0x305d2f0 <line:69:10, col:22>
|   | | `-VarDecl 0x305d260 <col:10, col:21> col:17 used i 'size_t':'unsigned long' cinit
|   | |   `-ImplicitCastExpr 0x305d2d8 <col:21> 'size_t':'unsigned long' <IntegralCast>
|   | |     `-IntegerLiteral 0x305d2b8 <col:21> 'int' 0
|   | |-<<<NULL>>>
|   | |-BinaryOperator 0x305d388 <col:24, col:28> '_Bool' '<'
|   | | |-ImplicitCastExpr 0x305d358 <col:24> 'size_t':'unsigned long' <LValueToRValue>
|   | | | `-DeclRefExpr 0x305d308 <col:24> 'size_t':'unsigned long' lvalue Var 0x305d260 'i' 'size_t':'unsigned long'
|   | | `-ImplicitCastExpr 0x305d370 <col:28> 'size_t':'unsigned long' <LValueToRValue>
|   | |   `-DeclRefExpr 0x305d330 <col:28> 'const size_t':'const unsigned long' lvalue Var 0x305ae90 'ELEMS' 'const size_t':'const unsigned long'
|   | |-UnaryOperator 0x305d3d8 <col:35, col:37> 'size_t':'unsigned long' lvalue prefix '++'
|   | | `-DeclRefExpr 0x305d3b0 <col:37> 'size_t':'unsigned long' lvalue Var 0x305d260 'i' 'size_t':'unsigned long'
|   | `-IfStmt 0x305d728 <line:70:9, line:71:38>
|   |   |-<<<NULL>>>
|   |   |-UnaryOperator 0x305d4d0 <line:70:13, col:20> '_Bool' prefix '!'
|   |   | `-ImplicitCastExpr 0x305d4b8 <col:14, col:20> '_Bool' <IntegralToBoolean>
|   |   |   `-ParenExpr 0x305d498 <col:14, col:20> 'unsigned long'
|   |   |     `-BinaryOperator 0x305d470 <col:15, col:19> 'unsigned long' '%'
|   |   |       |-ImplicitCastExpr 0x305d440 <col:15> 'size_t':'unsigned long' <LValueToRValue>
|   |   |       | `-DeclRefExpr 0x305d3f8 <col:15> 'size_t':'unsigned long' lvalue Var 0x305d260 'i' 'size_t':'unsigned long'
|   |   |       `-ImplicitCastExpr 0x305d458 <col:19> 'unsigned long' <IntegralCast>
|   |   |         `-IntegerLiteral 0x305d420 <col:19> 'int' 2
|   |   |-BinaryOperator 0x305d700 <line:71:13, col:38> 'float' lvalue '='
|   |   | |-DeclRefExpr 0x305d4f0 <col:13> 'float' lvalue Var 0x305ba60 'k' 'float'
|   |   | `-BinaryOperator 0x305d6d8 <col:17, col:38> 'float' '+'
|   |   |   |-ImplicitCastExpr 0x305d6a8 <col:17> 'float' <LValueToRValue>
|   |   |   | `-DeclRefExpr 0x305d518 <col:17> 'float' lvalue Var 0x305ba60 'k' 'float'
|   |   |   `-ImplicitCastExpr 0x305d6c0 <col:21, col:38> 'float' <LValueToRValue>
|   |   |     `-ArraySubscriptExpr 0x305d680 <col:21, col:38> 'float' lvalue
|   |   |       |-ImplicitCastExpr 0x305d668 <col:21> 'float *' <ArrayToPointerDecay>
|   |   |       | `-DeclRefExpr 0x305d540 <col:21> 'float [1000000]' lvalue Var 0x305bf00 'cum_sum' 'float [1000000]'
|   |   |       `-BinaryOperator 0x305d640 <col:29, col:37> 'unsigned long' '-'
|   |   |         |-BinaryOperator 0x305d5e0 <col:29, col:33> 'unsigned long' '+'
|   |   |         | |-ImplicitCastExpr 0x305d5b0 <col:29> 'size_t':'unsigned long' <LValueToRValue>
|   |   |         | | `-DeclRefExpr 0x305d568 <col:29> 'size_t':'unsigned long' lvalue Var 0x305d260 'i' 'size_t':'unsigned long'
|   |   |         | `-ImplicitCastExpr 0x305d5c8 <col:33> 'unsigned long' <IntegralCast>
|   |   |         |   `-IntegerLiteral 0x305d590 <col:33> 'int' 1
|   |   |         `-ImplicitCastExpr 0x305d628 <col:37> 'unsigned long' <IntegralCast>
|   |   |           `-IntegerLiteral 0x305d608 <col:37> 'int' 1
|   |   `-<<<NULL>>>
|   `-ReturnStmt 0x305d7f0 <line:73:5, col:12>
|     `-ImplicitCastExpr 0x305d7d8 <col:12> 'int' <FloatingToIntegral>
|       `-ImplicitCastExpr 0x305d7c0 <col:12> 'float' <LValueToRValue>
|         `-DeclRefExpr 0x305d798 <col:12> 'float' lvalue Var 0x305ba60 'k' 'float'
`-FunctionDecl 0x305daf0 prev 0x305ad00 <line:76:1, line:81:1> line:76:6 used mmult 'void (float **, float **, float **, size_t)'
  |-ParmVarDecl 0x305d8e0 <col:12, col:20> col:20 used A 'float **'
  |-ParmVarDecl 0x305d970 <col:23, col:31> col:31 used B 'float **'
  |-ParmVarDecl 0x305d9e0 <col:34, col:42> col:42 used C 'float **'
  |-ParmVarDecl 0x305da50 <col:45, col:52> col:52 used dim 'size_t':'unsigned long'
  `-CompoundStmt 0x305e580 <col:56, line:81:1>
    `-ForStmt 0x305e540 <line:77:5, line:80:44>
      |-DeclStmt 0x305dc50 <line:77:10, col:22>
      | `-VarDecl 0x305dbc0 <col:10, col:21> col:17 used i 'size_t':'unsigned long' cinit
      |   `-ImplicitCastExpr 0x305dc38 <col:21> 'size_t':'unsigned long' <IntegralCast>
      |     `-IntegerLiteral 0x305dc18 <col:21> 'int' 0
      |-<<<NULL>>>
      |-BinaryOperator 0x305dce8 <col:24, col:28> '_Bool' '<'
      | |-ImplicitCastExpr 0x305dcb8 <col:24> 'size_t':'unsigned long' <LValueToRValue>
      | | `-DeclRefExpr 0x305dc68 <col:24> 'size_t':'unsigned long' lvalue Var 0x305dbc0 'i' 'size_t':'unsigned long'
      | `-ImplicitCastExpr 0x305dcd0 <col:28> 'size_t':'unsigned long' <LValueToRValue>
      |   `-DeclRefExpr 0x305dc90 <col:28> 'size_t':'unsigned long' lvalue ParmVar 0x305da50 'dim' 'size_t':'unsigned long'
      |-UnaryOperator 0x305dd38 <col:33, col:35> 'size_t':'unsigned long' lvalue prefix '++'
      | `-DeclRefExpr 0x305dd10 <col:35> 'size_t':'unsigned long' lvalue Var 0x305dbc0 'i' 'size_t':'unsigned long'
      `-ForStmt 0x305e500 <line:78:9, line:80:44>
        |-DeclStmt 0x305de00 <line:78:14, col:26>
        | `-VarDecl 0x305dd70 <col:14, col:25> col:21 used j 'size_t':'unsigned long' cinit
        |   `-ImplicitCastExpr 0x305dde8 <col:25> 'size_t':'unsigned long' <IntegralCast>
        |     `-IntegerLiteral 0x305ddc8 <col:25> 'int' 0
        |-<<<NULL>>>
        |-BinaryOperator 0x305de98 <col:28, col:32> '_Bool' '<'
        | |-ImplicitCastExpr 0x305de68 <col:28> 'size_t':'unsigned long' <LValueToRValue>
        | | `-DeclRefExpr 0x305de18 <col:28> 'size_t':'unsigned long' lvalue Var 0x305dd70 'j' 'size_t':'unsigned long'
        | `-ImplicitCastExpr 0x305de80 <col:32> 'size_t':'unsigned long' <LValueToRValue>
        |   `-DeclRefExpr 0x305de40 <col:32> 'size_t':'unsigned long' lvalue ParmVar 0x305da50 'dim' 'size_t':'unsigned long'
        |-UnaryOperator 0x305dee8 <col:37, col:39> 'size_t':'unsigned long' lvalue prefix '++'
        | `-DeclRefExpr 0x305dec0 <col:39> 'size_t':'unsigned long' lvalue Var 0x305dbc0 'i' 'size_t':'unsigned long'
        `-ForStmt 0x305e4c0 <line:79:13, line:80:44>
          |-DeclStmt 0x305dfb0 <line:79:18, col:30>
          | `-VarDecl 0x305df20 <col:18, col:29> col:25 used k 'size_t':'unsigned long' cinit
          |   `-ImplicitCastExpr 0x305df98 <col:29> 'size_t':'unsigned long' <IntegralCast>
          |     `-IntegerLiteral 0x305df78 <col:29> 'int' 0
          |-<<<NULL>>>
          |-BinaryOperator 0x305e048 <col:32, col:36> '_Bool' '<'
          | |-ImplicitCastExpr 0x305e018 <col:32> 'size_t':'unsigned long' <LValueToRValue>
          | | `-DeclRefExpr 0x305dfc8 <col:32> 'size_t':'unsigned long' lvalue Var 0x305df20 'k' 'size_t':'unsigned long'
          | `-ImplicitCastExpr 0x305e030 <col:36> 'size_t':'unsigned long' <LValueToRValue>
```

```
|     `-DeclRefExpr 0x305dff0 <col:36> 'size_t':'unsigned long' lvalue ParmVar 0x305da50 'dim' 'size_t':'unsigned long'
|-UnaryOperator 0x305e098 <col:41, col:43> 'size_t':'unsigned long' lvalue prefix '++'
| `-DeclRefExpr 0x305e070 <col:43> 'size_t':'unsigned long' lvalue Var 0x305dbc0 'i' 'size_t':'unsigned long'
`-CompoundAssignOperator 0x305e488 <line:80:17, col:44> 'float' lvalue '+=' ComputeLHSTy='float' ComputeResultTy='float'
  |-ArraySubscriptExpr 0x305e1b8 <col:17, col:23> 'float' lvalue
  | |-ImplicitCastExpr 0x305e188 <col:17, col:20> 'float *' <LValueToRValue>
  | | `-ArraySubscriptExpr 0x305e138 <col:17, col:20> 'float *' lvalue
  | |   |-ImplicitCastExpr 0x305e108 <col:17> 'float **' <LValueToRValue>
  | |   | `-DeclRefExpr 0x305e0b8 <col:17> 'float **' lvalue ParmVar 0x305d9e0 'C' 'float **'
  | |   `-ImplicitCastExpr 0x305e120 <col:19> 'size_t':'unsigned long' <LValueToRValue>
  | |     `-DeclRefExpr 0x305e0e0 <col:19> 'size_t':'unsigned long' lvalue Var 0x305dbc0 'i' 'size_t':'unsigned long'
  | `-ImplicitCastExpr 0x305e1a0 <col:22> 'size_t':'unsigned long' <LValueToRValue>
  |   `-DeclRefExpr 0x305e160 <col:22> 'size_t':'unsigned long' lvalue Var 0x305dd70 'j' 'size_t':'unsigned long'
  `-BinaryOperator 0x305e460 <col:28, col:44> 'float' '*'
    |-ImplicitCastExpr 0x305e430 <col:28, col:34> 'float' <LValueToRValue>
    | `-ArraySubscriptExpr 0x305e2e0 <col:28, col:34> 'float' lvalue
    |   |-ImplicitCastExpr 0x305e2b0 <col:28, col:31> 'float *' <LValueToRValue>
    |   | `-ArraySubscriptExpr 0x305e260 <col:28, col:31> 'float *' lvalue
    |   |   |-ImplicitCastExpr 0x305e230 <col:28> 'float **' <LValueToRValue>
    |   |   | `-DeclRefExpr 0x305e1e0 <col:28> 'float **' lvalue ParmVar 0x305d8e0 'A' 'float **'
    |   |   `-ImplicitCastExpr 0x305e248 <col:30> 'size_t':'unsigned long' <LValueToRValue>
    |   |     `-DeclRefExpr 0x305e208 <col:30> 'size_t':'unsigned long' lvalue Var 0x305dbc0 'i' 'size_t':'unsigned long'
    |   `-ImplicitCastExpr 0x305e2c8 <col:33> 'size_t':'unsigned long' <LValueToRValue>
    |     `-DeclRefExpr 0x305e288 <col:33> 'size_t':'unsigned long' lvalue Var 0x305df20 'k' 'size_t':'unsigned long'
    `-ImplicitCastExpr 0x305e448 <col:38, col:44> 'float' <LValueToRValue>
      `-ArraySubscriptExpr 0x305e408 <col:38, col:44> 'float' lvalue
        |-ImplicitCastExpr 0x305e3d8 <col:38, col:41> 'float *' <LValueToRValue>
        | `-ArraySubscriptExpr 0x305e388 <col:38, col:41> 'float *' lvalue
        |   |-ImplicitCastExpr 0x305e358 <col:38> 'float **' <LValueToRValue>
        |   | `-DeclRefExpr 0x305e308 <col:38> 'float **' lvalue ParmVar 0x305d970 'B' 'float **'
        |   `-ImplicitCastExpr 0x305e370 <col:40> 'size_t':'unsigned long' <LValueToRValue>
        |     `-DeclRefExpr 0x305e330 <col:40> 'size_t':'unsigned long' lvalue Var 0x305df20 'k' 'size_t':'unsigned long'
        `-ImplicitCastExpr 0x305e3f0 <col:43> 'size_t':'unsigned long' <LValueToRValue>
          `-DeclRefExpr 0x305e3b0 <col:43> 'size_t':'unsigned long' lvalue Var 0x305dd70 'j' 'size_t':'unsigned long'
```