```
%9:
                    %10 = \text{sext i} 32 \% 4 \text{ to i} 64
                    %11 = icmp slt i64 %10, 256
                    %12 = select i1 %11, i64 %10, i64 256
                    %mul.i.i = shl i64 %6, 8
                    %sub.i = sub i32 -2, %3
                    %sub2.i = add i32 %sub.i, %4, !llvm.access.group !12
                    %mul.i = mul nsw i32 %sub2.i, %4, !llvm.access.group !12
                    %sub3.i = sub i32 -3, %3
                    %sub4.i = add i32 %sub3.i, %4, !llvm.access.group !12
                    %mul5.i = mul nsw i32 %sub4.i, %4, !llvm.access.group !12
                    %13 = icmp ugt i64 %12, 1
                    %umax = select i1 %13, i64 %12, i64 1
                    %min.iters.check = icmp ult i64 %umax, 8
                    br i1 %min.iters.check, label %pregion for entry.entry.i.preheader, label
                    ... %vector.scevcheck
                                     Τ
                                                                           F
                                         vector.scevcheck:
                                          %14 = add nsw i64 %umax, -1
                                          %15 = mul i32 %sub2.i, %4
                                          %16 = trunc i64 %6 to i32
                                          %17 = \text{shl i} 32 \%16, 8
                                          %18 = add nsw i32 %15, %17
                                          %19 = trunc i64 %14 to i32
                                          %20 = add i32 %18, %19
                                          %21 = icmp slt i32 %20, %18
                                          %22 = icmp ugt i64 %14, 4294967295
                                          %23 = \text{ or i } 1 \%21, \%22
                                          %24 = mul i32 %sub4.i, %4
                                          %25 = add nsw i32 %24, %17
                                          %26 = trunc i64 %14 to i32
                                          %27 = add i32 \%25, \%26
                                          %28 = icmp slt i32 \%27, \%25
                                         %29 = icmp ugt i64 %14, 4294967295
                                          %30 = \text{ or i } 1 \%28, \%29
                                          %31 = \text{ or i } 1 \%23, \%30
                                          br i1 %31, label %pregion for entry.entry.i.preheader, label %vector.ph
                                                                                                F
                                                        vector.ph:
                                                         %n.vec = and i64 %umax, -8
                                                         %broadcast.splatinsert = insertelement <8 x i64> undef, i64 %mul.i.i, i32 0
                                                        %broadcast.splat = shufflevector <8 x i64> %broadcast.splatinsert, <8 x i64>
                                                        ... undef, <8 x i32> zeroinitializer
                                                         %broadcast.splatinsert4 = insertelement <8 x i32> undef, i32 %mul.i, i32 0
                                                        %broadcast.splat5 = shufflevector <8 x i32> %broadcast.splatinsert4, <8 x
                                                        ... i32> undef, <8 x i32> zeroinitializer
                                                         %broadcast.splatinsert6 = insertelement <8 x i32> undef, i32 %mul5.i, i32 0
                                                         %broadcast.splat7 = shufflevector <8 x i32> %broadcast.splatinsert6, <8 x
                                                        ... i32> undef, <8 x i32> zeroinitializer
                                                        br label %vector.body
                                                     vector.body:
                                                     %index = phi i64 [ 0, %vector.ph ], [ %index.next, %vector.body ]
                                                     %vec.ind = phi <8 x i64> [ <i64 0, i64 1, i64 2, i64 3, i64 4, i64 5, i64 6,
                                                     ... i64 7>, %vector.ph ], [ %vec.ind.next, %vector.body ]
                                                     %32 = add <8 x i64> %vec.ind, %broadcast.splat, !llvm.access.group !12
                                                     %33 = trunc < 8 \times i64 > %32 \text{ to } < 8 \times i32 >, !llvm.access.group !12
                                                     %34 = add nsw <8 x i32> %broadcast.splat5, %33, !llvm.access.group !12
                                                     %35 = \text{sext} < 8 \times i32 > \%34 \text{ to } < 8 \times i64 >, !llvm.access.group !12
                                                     %36 = getelementptr inbounds float, float* %2, <8 x i64> %35,
                                                     ...!llvm.access.group!12
                                                     %wide.masked.gather = call <8 x float> @llvm.masked.gather.v8f32.v8p0f32(<8
                                                     ... x float*> %36, i32 4, <8 x i1> <i1 true, i1 true, i1 true, i1 true, i1 true,
                                                     ... i1 true, i1 true, i1 true>, <8 x float> undef), !tbaa !14, !llvm.access.group
                                                     ... !12
                                                     %37 = add nsw <8 x i32> %broadcast.splat7, %33, !llvm.access.group !12
                                                     %38 = \text{sext} < 8 \times i32 > %37 \text{ to } < 8 \times i64 >, !llvm.access.group !12
                                                     %39 = getelementptr inbounds float, float* %2, <8 x i64> %38,
                                                     ...!llvm.access.group!12
                                                     %wide.masked.gather8 = call <8 x float> @llvm.masked.gather.v8f32.v8p0f32(<8
                                                     ... x float*> %39, i32 4, <8 x i1> <i1 true, i1 true, i1 true, i1 true, i1 true,
                                                     ... i1 true, i1 true, i1 true>, <8 x float> undef), !tbaa !14, !llvm.access.group
                                                     ... !12
                                                     \%40 = getelementptr inbounds float, float* \%0, <8 x i64> \%38,
                                                     ...!llvm.access.group!12
                                                     %wide.masked.gather9 = call <8 x float> @llvm.masked.gather.v8f32.v8p0f32(<8
                                                     ... x float*> %40, i32 4, <8 x i1> <i1 true, i1 true, i1 true, i1 true, i1 true,
                                                     ... i1 true, i1 true, i1 true>, <8 x float> undef), !tbaa !14, !llvm.access.group
                                                     ... !12
                                                     %41 = fneg <8 x float> %wide.masked.gather8, !llvm.access.group !12
                                                     %42 = call <8 x float> @llvm.fmuladd.v8f32(<8 x float> %41, <8 x float>
                                                     ... %wide.masked.gather9, <8 x float> %wide.masked.gather), !llvm.access.group !12
                                                     %43 = getelementptr inbounds float, float* %1, <8 x i64> %35,
                                                     ...!llvm.access.group!12
                                                     %wide.masked.gather10 = call <8 x float>
                                                     ... @llvm.masked.gather.v8f32.v8p0f32(< 8 \times float* > %43, i32 4, < 8 \times i1 > < i1 \text{ true},
                                                     ... i1 true, i1 true, i1 true, i1 true, i1 true, i1 true, i1 true>, <8 x float>
                                                     ... undef), !tbaa !14, !llvm.access.group !12
                                                     %44 = fdiv < 8 \times float > %42, %wide.masked.gather10, !fpmath !18,
                                                     ...!llvm.access.group!12
                                                     call void @llvm.masked.scatter.v8f32.v8p0f32(<8 x float> %44, <8 x float*>
                                                     ... %36, i32 4, <8 x i1> <i1 true, i1 true, i1 true, i1 true, i1 true, i1 true, i1 true,
                                                     ... i1 true, i1 true>), !tbaa !14, !llvm.access.group !12
                                                     %index.next = add i64 %index, 8
                                                     %vec.ind.next = add <8 x i64> %vec.ind, <i64 8, i64 8, i64 8, i64 8, i64 8,
                                                     ... i64 8, i64 8, i64 8>
                                                     %45 = icmp eq i64 %index.next, %n.vec
                                                     br i1 %45, label %middle.block, label %vector.body, !llvm.loop !19
                                                                         Τ
                                                                                                                       F
                                                    middle.block:
                                                    %cmp.n = icmp eq i64 %umax, %n.vec
                                                    br i1 %cmp.n, label %adi kernel6.exit, label
                                                    ... %pregion for entry.entry.i.preheader
   pregion_for_entry.entry.i.preheader:
   % local id x.0.ph = phi i64 [ 0, %vector.scevcheck ], [ 0, %9 ], [ %n.vec,
   ... \( \bar{\pi} \) middle_block 1
    br label %pregion for entry.entry.i
pregion for entry.entry.i:
% local id x.0 = phi i64 [ %51, %pregion_for_entry.entry.i ], [
... % local id x.0.ph, %pregion for entry.entry.i.preheader ]
%add1.i.i = add i64 % local id x.0, %mul.i.i, !llvm.access.group !12
%conv.i = trunc i64 %add1.i.i to i32, !llvm.access.group !12
%add.i = add nsw i32 %mul.i, %conv.i, !llvm.access.group !12
%idxprom.i = sext i32 %add.i to i64, !llvm.access.group !12
%arrayidx.i = getelementptr inbounds float, float* %2, i64 %idxprom.i,
..!llvm.access.group!12
%46 = load float, float* %arrayidx.i, align 4, !tbaa !14, !llvm.access.group
%add6.i = add nsw i32 %mul5.i, %conv.i, !llvm.access.group !12
%idxprom7.i = sext i32 %add6.i to i64, !llvm.access.group !12
%arrayidx8.i = getelementptr inbounds float, float* %2, i64 %idxprom7.i,
...!llvm.access.group!12
%47 = load float, float* %arrayidx8.i, align 4, !tbaa !14,
...!llvm.access.group!12
%arrayidx14.i = getelementptr inbounds float, float* %0, i64 %idxprom7.i,
...!llvm.access.group!12
%48 = load float, float* %arrayidx14.i, align 4, !tbaa !14,
...!llvm.access.group!12
%neg.i = fneg float %47, !llvm.access.group !12
%49 = tail call float @llvm.fmuladd.f32(float %neg.i, float %48, float %46)
... #5. !llvm.access.group !12
%arrayidx21.i = getelementptr inbounds float, float* %1, i64 %idxprom.i,
...!llvm.access.group!12
%50 = load float, float* %arrayidx21.i, align 4, !tbaa !14,
...!llvm.access.group!12
%div.i = fdiv float %49, %50, !fpmath !18, !llvm.access.group !12
store float %div.i, float* %arrayidx.i, align 4, !tbaa !14,
...!llvm.access.group!12
\%51 = add nuw i64\% local id x.0, 1
%exitcond.not = icmp eq i6\overline{4} %51, %umax
br i1 %exitcond.not, label %adi kernel6.exit.loopexit, label
... %pregion for entry.entry.i, !llvm.loop!22
                                                         F
                            adi kernel6.exit.loopexit:
                             br label %adi kernel6.exit
```

adi kernel6.exit:

ret void

... !12