```
%8:
                            \%9 = \text{sext i} 32 \%3 \text{ to i} 64
                            %10 = icmp slt i64 \%9, 256
                            %11 = select i1 %10, i64 %9, i64 256
                            %mul.i.i = shl i64 %5, 8
                            %sub.i = add nsw i32 %3, -1, !llvm.access.group !12
                            %mul.i = mul nsw i32 %sub.i, %3, !llvm.access.group !12
                            %12 = icmp ugt i64 %11, 1
                            %umax = select i1 %12, i64 %11, 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:
                                                           %13 = add nsw i64 %umax, -1
                                                           %14 = add i32 \%3, -1
                                                           %15 = mul i32 %14, %3
                                                           %16 = trunc i64 %5 to i32
                                                           %17 = \text{shl i} 32 \%16, 8
                                                           %18 = add nsw i32 %15, %17
                                                           %19 = \text{trunc } i64 \%13 \text{ to } i32
                                                           %20 = add i32 %18, %19
                                                           %21 = icmp slt i32 %20, %18
                                                           %22 = icmp ugt i64 %13, 4294967295
                                                           %23 = \text{ or i } 1 \% 21, \%22
                                                           br i1 %23, label %pregion for entry.entry.i.preheader, label %vector.ph
                                                                                  Τ
                                                                                                                                         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.splatinsert1 = insertelement <8 x i32> undef, i32 %mul.i, i32 0
                                                                                %broadcast.splat2 = shufflevector <8 x i32> %broadcast.splatinsert1, <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 ]
                                                                              %24 = add <8 x i64> %vec.ind, %broadcast.splat, !llvm.access.group !12
                                                                              %25 = trunc <8 x i64> %24 to <8 x i32>, !llvm.access.group !12
                                                                              %26 = add nsw <8 x i32> %broadcast.splat2, %25, !llvm.access.group !12
                                                                              %27 = \text{sext} < 8 \times \text{i}32 > \%26 \text{ to} < 8 \times \text{i}64 >, !llvm.access.group !12
                                                                              \%28 = getelementptr inbounds float, float* \%2, <8 \times i64 > \%27,
                                                                              ...!llvm.access.group!12
                                                                              %wide.masked.gather = call <8 x float> @llvm.masked.gather.v8f32.v8p0f32(<8
                                                                             ... x float*> %28, 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
                                                                             %29 = getelementptr inbounds float, float* %1, <8 x i64> %27,
                                                                             ...!llvm.access.group!12
                                                                              %wide.masked.gather3 = call <8 x float> @llvm.masked.gather.v8f32.v8p0f32(<8
                                                                             ... x float*> %29, i32 4, <8 x i1> <i1 true, i1 
                                                                             ... i1 true, i1 true, i1 true, <8 x float> undef), !tbaa !14, !llvm.access.group
                                                                             ...!12
                                                                              %30 = fdiv <8 x float> %wide.masked.gather, %wide.masked.gather3, !fpmath
                                                                             ...!18,!llvm.access.group!12
                                                                             call void @llvm.masked.scatter.v8f32.v8p0f32(<8 x float> %30, <8 x float*>
                                                                             ... %28, 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>
                                                                              %31 = icmp eq i64 %index.next, %n.vec
                                                                              br i1 %31, label %middle.block, label %vector.body, !llvm.loop !19
                                                                                                                                                                         F
                                                                          middle.block:
                                                                           %cmp.n = icmp eq i64 %umax, %n.vec
                                                                           br i1 %cmp.n, label %adi kernel5.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, %8], [%n.vec, ]
    ... \mathfrak{\pi}middle.block \cdot\
    br label %pregion for entry.entry.i
pregion for entry.entry.i:
% local id \bar{x}.0 = phi i64 [ %34, %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
%32 = load float, float* %arrayidx.i, align 4, !tbaa !14, !llvm.access.group
%arrayidx6.i = getelementptr inbounds float, float* %1, i64 %idxprom.i,
...!llvm.access.group!12
%33 = load float, float* %arrayidx6.i, align 4, !tbaa !14,
...!llvm.access.group!12
%div.i = fdiv float %32, %33, !fpmath !18, !llvm.access.group !12
store float %div.i, float* %arrayidx.i, align 4, !tbaa !14,
...!llvm.access.group!12
%34 = add nuw i64 \% local id x.0, 1
%exitcond.not = icmp eq i6\overline{4} %34, %umax
br i1 %exitcond.not, label %adi_kernel5.exit.loopexit, label
... %pregion for entry.entry.i, !llvm.loop!22
                                                                                 F
                                        adi_kernel5.exit.loopexit:
                                        br label %adi kernel5.exit
                                                            adi kernel5.exit:
```

ret void

... !12