```
%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
                            %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:
                                                           %ident.check = icmp ne i32 %3, 1
                                                           %13 = add nsw i64 %umax, -1
                                                           %14 = \text{trunc } i64 \%5 \text{ to } i32
                                                           %15 = \text{shl i} 32 \%14, 8
                                                           %16 = \text{trunc } i64 \%13 \text{ to } i32
                                                           %17 = add i32 %15, %16
                                                          %18 = icmp slt i32 %17, %15
                                                           %19 = icmp ugt i64 %13, 4294967295
                                                          %20 = \text{ or i } 1 \% 18, \% 19
                                                           %21 = or i1 %ident.check, %20
                                                          br i1 %21, label %pregion for entry.entry.i.preheader, label %vector.ph
                                                                               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 %3, i32 0
                                                                                %broadcast.splat2 = shufflevector <8 x i32> %broadcast.splatinsert1, <8 x
                                                                                ... i32> undef, <8 x i32> zeroinitializer
                                                                                %broadcast.splatinsert3 = insertelement <8 x i32> undef, i32 %sub.i, i32 0
                                                                                %broadcast.splat4 = shufflevector <8 x i32> %broadcast.splatinsert3, <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 ]
                                                                             %22 = add <8 x i64> %vec.ind, %broadcast.splat, !llvm.access.group !12
                                                                             \%23 = \text{trunc} < 8 \times i64 > \%22 \text{ to } < 8 \times i32 >, !llvm.access.group !12
                                                                             %24 = mul nsw <8 x i32> %broadcast.splat2, %23, !llvm.access.group !12
                                                                             %25 = add nsw <8 x i32> %broadcast.splat4, %24, !llvm.access.group !12
                                                                             \%26 = \text{sext} < 8 \times i32 > \%25 \text{ to } < 8 \times i64 >, !llvm.access.group !12
                                                                             \%27 = getelementptr inbounds float, float* \%2, <8 x i64> \%26,
                                                                             ...!llvm.access.group!12
                                                                             %wide.masked.gather = call <8 x float> @llvm.masked.gather.v8f32.v8p0f32(<8
                                                                            ... x float*> %27, 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
                                                                             %28 = getelementptr inbounds float, float* %1, <8 x i64> %26,
                                                                             ...!llvm.access.group!12
                                                                             %wide.masked.gather5 = call <8 x float> @llvm.masked.gather.v8f32.v8p0f32(<8
                                                                            ... x float*> %28, i32 4, <8 x i1> <i1 true, i1 
                                                                            ... i1 true, i1 true, i1 true, <8 x float> undef), !tbaa !14, !llvm.access.group
                                                                            ...!12
                                                                             %29 = fdiv <8 x float> %wide.masked.gather, %wide.masked.gather5, !fpmath
                                                                            ...!18,!llvm.access.group!12
                                                                             call void @llvm.masked.scatter.v8f32.v8p0f32(<8 x float> %29, <8 x float*>
                                                                            ... %27, 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>
                                                                             %30 = icmp eq i64 %index.next, %n.vec
                                                                             br i1 %30, label %middle.block, label %vector.body, !llvm.loop !19
                                                                         middle.block:
                                                                          %cmp.n = icmp eq i64 %umax, %n.vec
                                                                          br i1 %cmp.n, label %adi kernel2.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,
    ... %middle.block 1
    br label %pregion for entry.entry.i
pregion for entry.entry.i:
% local id \bar{x}.0 = phi i64 [ %33, %pregion for entry.entry.i ], [
... \(\frac{1}{2}\) local_id_x.0.ph, \(\partial \) 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
%mul.i = mul nsw i32 %conv.i, %3, !llvm.access.group !12
%add.i = add nsw i32 %sub.i, %mul.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
%31 = 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
%32 = load float, float* %arrayidx6.i, align 4, !tbaa !14,
...!llvm.access.group!12
%div.i = fdiv float %31, %32, !fpmath !18, !llvm.access.group !12
store float %div.i, float* %arrayidx.i, align 4, !tbaa !14,
...!llvm.access.group!12
%33 = add nuw i64 \% local id x.0, 1
%exitcond.not = icmp eq i6\overline{4} %33, %umax
br i1 %exitcond.not, label %adi kernel2.exit.loopexit, label
... %pregion for entry.entry.i, !llvm.loop!22
                        Т
                                                                                 F
                                        adi kernel2.exit.loopexit:
                                        br label %adi kernel2.exit
                                                            adi kernel2.exit:
                                                            ret void
```

...!12