```
%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
%mul.i = mul nsw i32 %sub2.i, %4
%sub3.i = sub i32 -3, %3
%sub4.i = add i32 %sub3.i, %4
%mul5.i = mul nsw i32 %sub4.i, %4
%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
                                                          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 } i1 \%21, \%22
                     %24 = mul i32 %sub4.i, %4
                     %25 = add nsw i32 %24, %17
                     \%26 = \text{trunc } i64 \%14 \text{ 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.memcheck
                                                                                    F
                                              vector.memcheck:
                                               %32 = mul i32 %sub2.i, %4
                                               %33 = \text{sext i} 32 \% 32 \text{ to i} 64
                                               %34 = \text{trunc } i64 \%6 \text{ to } i32
                                               %35 = \text{shl i} 32 \%34, 8
                                               %36 = \text{sext i} 32 \% 35 \text{ to i} 64
                                               %37 = add nsw i64 %33, %36
                                               %scevgep = getelementptr float, float* %2, i64 %37
                                               %38 = add nsw i64 %umax, %33
                                               %39 = add nsw i64 %38, %36
                                               %scevgep5 = getelementptr float, float* %2, i64 %39
                                               %40 = mul i32 %sub4.i, %4
                                               %41 = \text{sext i} 32 \% 40 \text{ to i} 64
                                               %42 = add nsw i64 %41, %36
                                               %scevgep7 = getelementptr float, float* %2, i64 %42
                                               %43 = add nsw i64 %umax, %41
                                               %44 = add nsw i64 %43, %36
                                               %scevgep9 = getelementptr float, float* %2, i64 %44
                                               %scevgep11 = getelementptr float, float* %0, i64 %42
                                               %scevgep13 = getelementptr float, float* %0, i64 %44
                                               %scevgep15 = getelementptr float, float* %1, i64 %37
                                               %scevgep17 = getelementptr float, float* %1, i64 %39
                                               %bound0 = icmp ult float* %scevgep, %scevgep9
                                               %bound1 = icmp ult float* %scevgep7, %scevgep5
                                               %found.conflict = and i1 %bound0, %bound1
                                               %bound019 = icmp ult float* %scevgep, %scevgep13
%bound120 = icmp ult float* %scevgep11, %scevgep5
                                               %found.conflict21 = and i1 %bound019, %bound120
                                               %conflict.rdx = or i1 %found.conflict, %found.conflict21
                                               %bound022 = icmp ult float* %scevgep, %scevgep17
%bound123 = icmp ult float* %scevgep15, %scevgep5
                                               %found.conflict24 = and i1 %bound022, %bound123
                                               %conflict.rdx25 = or i1 %conflict.rdx, %found.conflict24
                                               br i1 %conflict.rdx25, label %pregion for entry.entry.i.preheader, label
                                               ... %vector.ph
                                                                Τ
                                                                                                       F
                                                                                         vector.ph:
                                                                                         %n.vec = and i64 %umax, -8
                                                                                         br label %vector.body
                                                                vector.body:
                                                                %index = phi i64 [ 0, %vector.ph ], [ %index.next, %vector.body ]
                                                                %45 = add i64 %index, %mul.i.i
                                                                %46 = \text{trunc } i64 \%45 \text{ to } i32
                                                                %47 = add nsw i32 %mul.i, %46
                                                                %48 = \text{sext i} 32 \% 47 \text{ to i} 64
                                                                %49 = getelementptr inbounds float, float* %2, i64 %48
                                                                %50 = bitcast float* %49 to <8 x float>*
                                                                %wide.load = load <8 x float>, <8 x float>* %50, align 4, !tbaa !12,
                                                                ... !alias.scope !16, !noalias !19
                                                                %51 = add nsw i32 %mul5.i, %46
                                                                \%52 = \text{sext i} 32 \% 51 \text{ to i} 64
                                                                %53 = getelementptr inbounds float, float* %2, i64 %52
                                                                \%54 = bitcast float* \%53 to <8 x float>*
                                                                %wide.load26 = load <8 x float>, <8 x float>* %54, align 4, !tbaa !12,
                                                                ... !alias.scope !23
                                                                %55 = getelementptr inbounds float, float* %0, i64 %52
                                                                \%56 = bitcast float* \%55 to < 8 x float>*
                                                                %wide.load27 = load <8 x float>, <8 x float>* %56, align 4, !tbaa !12,
                                                                ... !alias.scope !24
                                                                %57 = \text{fneg} < 8 \text{ x float} > \% \text{wide.load26}
                                                                \%58 = \text{call} < 8 \times \text{float} > \text{@llvm.fmuladd.v8f32} (< 8 \times \text{float} > \%57, < 8 \times \text{float} > \text{grade})
                                                                ... %wide.load27, <8 x float> %wide.load)
                                                                %59 = getelementptr inbounds float, float* %1, i64 %48
                                                                \%60 = \text{bitcast float*} \%59 \text{ to } < 8 \text{ x float>*}
                                                                %wide.load28 = load <8 x float>, <8 x float>* %60, align 4, !tbaa !12,
                                                                ... !alias.scope !25
                                                                %61 = fdiv <8 x float> %58, %wide.load28, !fpmath !26
                                                                \%62 = bitcast float* \%49 to <8 x float>*
                                                                store <8 x float> %61, <8 x float>* %62, align 4, !tbaa !12, !alias.scope
                                                                ... !16, !noalias !19, !llvm.access.group !27
                                                                %index.next = add i64 %index, 8
                                                                %63 = icmp eq i64 %index.next, %n.vec
                                                                br i1 %63, label %middle.block, label %vector.body, !llvm.loop !29
                                                            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.memcheck ], [ 0, %vector.scevcheck
... ], [0, \( \sqrt{9} \)], [\( \sqrt{n.vec}, \sqrt{middle.block} \)]
br label %pregion for entry.entry.i
pregion for entry.entry.i:
% local id x.0 = phi i64 [ %69, %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
%conv.i = trunc i64 %add1.i.i to i32
%add.i = add nsw i32 %mul.i, %conv.i
%idxprom.i = sext i32 %add.i to i64
%arrayidx.i = getelementptr inbounds float, float* %2, i64 %idxprom.i %64 = load float, float* %arrayidx.i, align 4, !tbaa !12
%add6.i = add nsw i32 %mul5.i, %conv.i
%idxprom7.i = sext i32 %add6.i to i64
%arrayidx8.i = getelementptr inbounds float, float* %2, i64 %idxprom7.i
%65 = load float, float* %arrayidx8.i, align 4, !tbaa !12
%arrayidx14.i = getelementptr inbounds float, float* %0, i64 %idxprom7.i
%66 = load float, float* %arrayidx14.i, align 4, !tbaa !12
%neg.i = fneg float %65
%67 = tail call float @llvm.fmuladd.f32(float %neg.i, float %66, float %64)
... #3
%arrayidx21.i = getelementptr inbounds float, float* %1, i64 %idxprom.i
%68 = load float, float* %arrayidx21.i, align 4, !tbaa !12
%div.i = fdiv float %67, %68, !fpmath !26
store float %div.i, float* %arrayidx.i, align 4, !tbaa !12,
...!llvm.access.group!27
\%69 = \text{add nuw } i64 \% \text{ local id } x.0, 1
\%exitcond.not = icmp eq i6\overline{4} \%69, \%umax
br i1 %exitcond.not, label %adi kernel6.exit.loopexit, label
... %pregion for entry.entry.i, !llvm.loop!32
                             adi kernel6.exit.loopexit:
                              br label %adi kernel6.exit
```

adi kernel6.exit:

ret void