```
vector.scevcheck:
                                                        %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
                                                         %9 = mul i32 %sub2.i, %4
                                                         %10 = trunc i64 %6 to i32
                                                         %11 = shl i32 %10, 8
                                                         %12 = add i32 \%9, \%11
                                                         %13 = icmp sgt i32 %12, 2147483392
                                                         %14 = mul i32 %sub4.i, %4
                                                         %15 = add i32 %14, %11
                                                         %16 = icmp sgt i32 %15, 2147483392
                                                         %17 = \text{ or } i1 \%13, \%16
                                                         br i1 %17, label %pregion for entry.entry.i.preheader, label %vector.memcheck
                                                                                 vector.memcheck:
                                                                                 %18 = mul i32 %sub2.i, %4
                                                                                  %19 = trunc i64 %6 to i32
                                                                                  %20 = \text{shl i} 32 \%19, 8
                                                                                  %21 = add i32 %18. %20
                                                                                  %22 = \text{sext i} 32 \% 21 \text{ to i} 64
                                                                                  %scevgep = getelementptr float, float* %2, i64 %22
                                                                                  %23 = add nsw i64 %22. 256
                                                                                  %scevgep5 = getelementptr float, float* %2, i64 %23
                                                                                  %24 = mul i32 %sub4.i, %4
                                                                                  %25 = add i32 %24, %20
                                                                                  \%26 = \text{sext i} 32 \%25 \text{ to i} 64
                                                                                  %scevgep7 = getelementptr float, float* %2, i64 %26
                                                                                  %27 = add nsw i64 %26, 256
                                                                                  %scevgep9 = getelementptr float, float* %2, i64 %27
                                                                                  %scevgep11 = getelementptr float, float* %0, i64 %26
                                                                                  %scevgep13 = getelementptr float, float* %0, i64 %27
                                                                                  %scevgep15 = getelementptr float, float* %1, i64 %22
                                                                                  %scevgep17 = getelementptr float, float* %1, i64 %23
                                                                                  %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
                                                                                                     vector.ph:
                                                                                                      %broadcast.splatinsert = insertelement <8 x i64> undef, i64 %mul.i.i, i32 0
                                                                                                      %broadcast.splat = shufflevector <8 x i64> %broadcast.splatinsert, <8 x i64>
                                                         pregion for entry.entry.i.preheader:
                                                                                                      ... undef, <8 x i32> zeroinitializer
                                                          br label %pregion for entry.entry.i
                                                                                                      %broadcast.splatinsert26 = insertelement <8 x i32> undef, i32 %4, i32 0
                                                                                                      %broadcast.splat27 = shufflevector <8 x i32> %broadcast.splatinsert26, <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 ]
                                                                                                        %28 = add nuw nsw <8 x i64> %vec.ind, %broadcast.splat
                                                                                                       %29 = trunc < 8 \times i64 > %28 to < 8 \times i32 >
                                                                                                        %30 = icmp sgt <8 x i32> %broadcast.splat27, %29
                                                                                                       %31 = extractelement <8 x i32> %29, i32 0
                                                                                                        %32 = add nsw i32 %mul.i, %31
                                                                                                       %33 = \text{sext i} 32 \% 32 \text{ to i} 64
                                                                                                        %34 = getelementptr inbounds float, float* %2, i64 %33
                                                                                                       %35 = bitcast float* %34 to <8 x float>*
                                                                                                        %wide.masked.load = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x
                                                                                                        ... float>* %35, i32 4, <8 x i1> %30, <8 x float> undef), !tbaa !12, !alias.scope
                                                                                                        ... !16, !noalias !19
                                                                                                       %36 = add nsw i32 %mul5.i, %31
                                                                                                        %37 = \text{sext i} 32 \% 36 \text{ to i} 64
                                                                                                       %38 = getelementptr inbounds float, float* %2, i64 %37
                                                                                                       %39 = bitcast float* %38 to <8 x float>*
                                                                                                        %wide.masked.load28 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x
                                            pregion for entry.entry.i:
                                                                                                       ... float>* %39, i32 4, <8 x i1> %30, <8 x float> undef), !tbaa !12, !alias.scope
                                            % local id x.0 = phi i64 [ %54, %if.end.r exit.i ], [ 0,
                                                                                                        ... !23
                                            ... %pregion_for_entry.entry.i.preheader ]
                                                                                                       %40 = getelementptr inbounds float, float* %0, i64 %37
                                            %add1.i.i = add nuw nsw i64 % local id x.0, %mul.i.i
                                                                                                        %41 = bitcast float* %40 to <8 x float>*
                                            %conv.i = trunc i64 %add1.i.i to i32
                                                                                                        %wide.masked.load29 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x
                                            %cmp.i = icmp slt i32 %conv.i, %4
                                                                                                       ... float>* %41, i32 4, <8 x i1> %30, <8 x float> undef), !tbaa !12, !alias.scope
                                            br i1 %cmp.i, label %if.then.i, label %if.end.r exit.i
                                                                                                       ... !24
                                                                                    F
                                                                                                        %42 = fneg <8 x float> %wide.masked.load28
                                                                                                        \%43 = \text{call} < 8 \times \text{float} > \text{@llvm.fmuladd.v8f32} (< 8 \times \text{float} > \%42, < 8 \times \text{float} >
                                                                                                        ... %wide.masked.load29, <8 x float> %wide.masked.load)
                                                                                                        %44 = getelementptr inbounds float, float* %1, i64 %33
                                                                                                       %45 = bitcast float* %44 to <8 x float>*
                                                                                                        %wide.masked.load30 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x
                                                                                                       ... float>* %45, i32 4, <8 x i1> %30, <8 x float> undef), !tbaa !12, !alias.scope
                                                                                                       ... !25
                                                                                                        %46 = fdiv <8 x float> %43, %wide.masked.load30, !fpmath !26
                                                                                                       %47 = bitcast float* %34 to <8 x float>*
                                                                                                       call void @llvm.masked.store.v8f32.p0v8f32(<8 x float> %46, <8 x float>*
                                                                                                        ... %47, i32 4, <8 x i1> %30), !tbaa !12, !alias.scope !16, !noalias !19,
                                                                                                        ...!llvm.access.group!27
                                                                                                        %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>
                                                                                                        %48 = icmp eq i64 %index.next, 256
                                                                                                       br i1 %48, label %adi kernel6.exit.loopexit32, label %vector.body,
                                                                                                        ... !llvm.loop !29
if.then.i:
%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
%49 = 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
%50 = load float, float* %arrayidx8.i, align 4, !tbaa !12
%arrayidx14.i = getelementptr inbounds float, float* %0, i64 %idxprom7.i
%51 = load float, float* %arrayidx14.i, align 4, !tbaa !12
%neg.i = fneg float %50
%52 = tail call float @llvm.fmuladd.f32(float %neg.i, float %51, float %49)
%arrayidx21.i = getelementptr inbounds float, float* %1, i64 %idxprom.i
%53 = load float, float* %arrayidx21.i, align 4, !tbaa !12
%div.i = fdiv float %52, %53, !fpmath !26
store float %div.i, float* %arrayidx.i, align 4, !tbaa !12,
...!llvm.access.group!27
br label %if.end.r exit.i
                                    if.end.r exit.i:
                                    \%54 = add nuw nsw i64 % local id x.0, 1
                                    %exitcond.not = icmp eq i\overline{6}4 %5\overline{4}, \overline{2}56
                                                                                                          adi kernel6.exit.loopexit32:
                                    br i1 %exitcond.not, label %adi kernel6.exit.loopexit, label
                                                                                                          br label %adi kernel6.exit
                                    ... %pregion for entry.entry.i, !llvm.loop !32
                                                            adi kernel6.exit.loopexit:
                                                            br label %adi kernel6.exit
                                                                          adi kernel6.exit:
                                                                          ret void
```