```
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
                                                                                                                                                    F
                                                                                                                     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
                                                                                                                     %broadcast.splatinsert26 = insertelement <8 x i32> undef, i32 %4, i32 0
                                                                     br label %pregion for entry.entry.i
                                                                                                                     %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.1, %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.1, %vector.body ]
                                                                                                                         %28 = add nuw nsw <8 x i64> %vec.ind, %broadcast.splat
                                                                                                                         %29 = trunc <8 x i64> %28 to <8 x i32>
                                                                                                                         %30 = icmp sgt <8 x i32> %broadcast.splat27, %29
                                                                                                                         %31 = \text{extractelement} < 8 \text{ x i} 32 > %29, i 32 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
                                                                                                                         ... float>* %39, i32 4, <8 x i1> %30, <8 x float> undef), !tbaa !12, !alias.scope
                                                                                                                         %40 = getelementptr inbounds float, float* %0, i64 %37
                                                                                                                         %41 = bitcast float* %40 to <8 x float>*
                                                                                                                         %wide.masked.load29 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x
                                                                                                                         ... float>* %41, i32 4, <8 x i1> %30, <8 x float> undef), !tbaa !12, !alias.scope
                                                                                                                         ...!24
                                                                                                                         %42 = fneg <8 x float> %wide.masked.load28
                                                                                                                         %43 = call <8 x float> @llvm.fmuladd.v8f32(<8 x float> %42, <8 x 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
                                                                                                                         %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,
                                      pregion for entry.entry.i:
                                                                                                                         ...!llvm.access.group!27
                                      % local id x.0 = phi i64 [0, %pregion_for_entry.entry.i.preheader], [ %80,
                                                                                                                         %vec.ind.next = add <8 x i64> %vec.ind, <i64 8, i64 8, i64 8, i64 8, i64 8,
                                      ... \overline{\%}if.end.r exit.i.1
                                                                                                                          .. i64 8, i64 8, i64 8>
                                      %add1.i.i = add nuw nsw i64 %_local_id_x.0, %mul.i.i
                                                                                                                         %48 = add nuw nsw <8 x i64> %vec.ind.next, %broadcast.splat
                                      %conv.i = trunc i64 %add1.i.i to i32
                                                                                                                         %49 = \text{trunc} < 8 \times i64 > %48 \text{ to } < 8 \times i32 > 
                                      %cmp.i = icmp slt i32 %conv.i, %4
                                                                                                                         \%50 = icmp \, sgt < 8 \, x \, i32 > \%broadcast.splat27, \%49
                                      br i1 %cmp.i, label %if.then.i, label %if.end.r_exit.i
                                                                                                                         %51 = \text{extractelement} < 8 \times i32 > %49, i32 0
                                                                                                F
                                                                                                                         %52 = add nsw i32 %mul.i, %51
                                                                                                                         %53 = \text{sext i} 32 \% 52 \text{ to i} 64
                                                                                                                         %54 = getelementptr inbounds float, float* %2, i64 %53
                                                                                                                         %55 = bitcast float* %54 to <8 x float>*
                                                                                                                         %wide.masked.load.1 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x
                                                                                                                         ... float>* %55, i32 4, <8 x i1> %50, <8 x float> undef), !tbaa !12, !alias.scope
                                                                                                                         ... !16, !noalias !19
                                                                                                                         %56 = add nsw i32 %mul5.i, %51
                                                                                                                         %57 = \text{sext i} 32 \% 56 \text{ to i} 64
                                                                                                                         %58 = getelementptr inbounds float, float* %2, i64 %57 %59 = bitcast float* %58 to <8 x float>*
                                                                                                                         %wide.masked.load28.1 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8
                                                                                                                         ... x float>* %59, i32 4, <8 x i1> %50, <8 x float> undef), !tbaa !12,
                                                                                                                         ... !alias.scope !23
                                                                                                                         %60 = getelementptr inbounds float, float* %0, i64 %57 %61 = bitcast float* %60 to <8 x float>*
                                                                                                                         %wide.masked.load29.1 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8
                                                                                                                         ... x float>* %61, i32 4, <8 x i1> %50, <8 x float> undef), !tbaa !12,
                                                                                                                         ... !alias.scope !24
                                                                                                                         \%62 = \text{fneg} < 8 \text{ x float} > \% \text{wide.masked.load} 28.1
                                                                                                                         %63 = call <8 x float> @llvm.fmuladd.v8f32(<8 x float> %62, <8 x float>
                                                                                                                         ... %wide.masked.load29.1, <8 x float> %wide.masked.load.1)
                                                                                                                         %64 = getelementptr inbounds float, float* %1, i64 %53
                                                                                                                         %65 = bitcast float* %64 to <8 x float>*
                                                                                                                         %wide.masked.load30.1 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8
                                                                                                                         ... x float>* %65, i32 4, <8 x i1> %50, <8 x float> undef), !tbaa !12,
                                                                                                                         ... !alias.scope !25
                                                                                                                         %66 = fdiv <8 x float> %63, %wide.masked.load30.1, !fpmath !26
                                                                                                                         \%67 = bitcast float* \%54 to <8 x float>*
                                                                                                                         call void @llvm.masked.store.v8f32.p0v8f32(<8 x float> %66, <8 x float>*
                                                                                                                         ... %67, i32 4, <8 x i1> %50), !tbaa !12, !alias.scope !16, !noalias !19,
                                                                                                                         ...!llvm.access.group!27
                                                                                                                         %index.next.1 = add nuw nsw i64 %index, 16
                                                                                                                         %vec.ind.next.1 = add <8 x i64> %vec.ind, <i64 16, i64 16, i64 16, i64 16,
                                                                                                                         ... i64 16, i64 16, i64 16, i64 16>
                                                                                                                         %68 = icmp eq i64 %index.next.1, 256
                                                                                                                         br i1 %68, 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
              %69 = 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
             %70 = load float, float* %arrayidx8.i, align 4, !tbaa !12
             %arrayidx14.i = getelementptr inbounds float, float* %0, i64 %idxprom7.i
             %71 = load float, float* %arrayidx14.i, align 4, !tbaa !12
             %neg.i = fneg float %70
              %72 = tail call float @llvm.fmuladd.f32(float %neg.i, float %71, float %69)
             %arrayidx21.i = getelementptr inbounds float, float* %1, i64 %idxprom.i %73 = load float, float* %arrayidx21.i, align 4, !tbaa !12
              %div.i = fdiv float %72, %73, !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:
                            \%74 = \text{ or } i64 \% \text{ local id } x.0, 1
                            %add1.i.i.1 = add nuw nsw i64 %74, %mul.i.i
                                                                                                                     adi kernel6.exit.loopexit32:
                           %conv.i.1 = trunc i64 %add1.i.i.1 to i32
                                                                                                                      br label %adi kernel6.exit
                           %cmp.i.1 = icmp slt i32 %conv.i.1, %4
                           br i1 %cmp.i.1, label %if.then.i.1, label %if.end.r exit.i.1
                                                                       F
if.then.i.1:
%add.i.1 = add nsw i32 %mul.i, %conv.i.1
%idxprom.i.1 = sext i32 %add.i.1 to i64
%arrayidx.i.1 = getelementptr inbounds float, float* %2, i64 %idxprom.i.1
%75 = load float, float* %arrayidx.i.1, align 4, !tbaa !12
%add6.i.1 = add nsw i32 %mul5.i, %conv.i.1
%idxprom7.i.1 = sext i32 %add6.i.1 to i64
%arrayidx8.i.1 = getelementptr inbounds float, float* %2, i64 %idxprom7.i.1 %76 = load float, float* %arrayidx8.i.1, align 4, !tbaa !12
%arrayidx14.i.1 = getelementptr inbounds float, float* %0, i64 %idxprom7.i.1
%77 = load float, float* %arrayidx14.i.1, align 4, !tbaa !12
%neg.i.1 = fneg float %76
%78 = tail call float @llvm.fmuladd.f32(float %neg.i.1, float %77, float
.. %75) #5
%arrayidx21.i.1 = getelementptr inbounds float, float* %1, i64 %idxprom.i.1
%79 = load float, float* %arrayidx21.i.1, align 4, !tbaa !12
%div.i.1 = fdiv float %78, %79, !fpmath !26
store float %div.i.1, float* %arrayidx.i.1, align 4, !tbaa !12,
..!llvm.access.group!27
br label %if.end.r exit.i.1
                                                   if.end.r exit.i.1:
                                                    \%80 = \text{add nuw nsw } i64 \% \text{ local id } x.0, 2
                                                    %exitcond.not.1 = icmp eq i64 \sqrt{80}, 256
                                                   br i1 %exitcond.not.1, 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
                                                                                 CFG for 'pocl kernel adi kernel6' function
```