```
vector.scevcheck:
                                                                       \%9 = \text{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
                                                                       %10 = mul i32 %sub2.i, %4
                                                                       %11 = trunc i64 %6 to i32
                                                                       %12 = shl i32 %11, 8
                                                                       %13 = add i32 %10, %12
                                                                       %14 = icmp sgt i32 %13, 2147483392
                                                                       %15 = mul i32 %sub4.i, %4
                                                                       %16 = add i32 %15, %12
                                                                       %17 = icmp sgt i32 %16, 2147483392
                                                                       %18 = \text{ or } i1 \%14, \%17
                                                                       br i1 %18, label %pregion for entry.entry.i.preheader, label %vector.memcheck
                                                                                                  vector.memcheck:
                                                                                                  %19 = mul i32 %sub2.i. %4
                                                                                                   %20 = trunc i64 %6 to i32
                                                                                                   %21 = shl i32 %20, 8
                                                                                                   %22 = add i32 %19, %21
                                                                                                   %23 = \text{sext i} 32 \% 22 \text{ to i} 64
                                                                                                   %scevgep = getelementptr float, float* %2, i64 %23
                                                                                                   %24 = add nsw i64 %23. 256
                                                                                                   %scevgep5 = getelementptr float, float* %2, i64 %24
                                                                                                   %25 = mul i32 %sub4.i, %4
                                                                                                   %26 = add i32 %25, %21
                                                                                                   %27 = \text{sext i} 32 \% 26 \text{ to i} 64
                                                                                                   %scevgep7 = getelementptr float, float* %2, i64 %27
                                                                                                   %28 = add nsw i64 %27, 256
                                                                                                   %scevgep9 = getelementptr float, float* %2, i64 %28
                                                                                                   %scevgep11 = getelementptr float, float* %0, i64 %27
                                                                                                   %scevgep13 = getelementptr float, float* %0, i64 %28
                                                                                                   %scevgep15 = getelementptr float, float* %1, i64 %23
                                                                                                   %scevgep17 = getelementptr float, float* %1, i64 %24
                                                                                                   %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 %9, i32 0
                                                                                                                           %broadcast.splat = shufflevector <8 x i64> %broadcast.splatinsert, <8 x i64>
                                                                        pregion for entry.entry.i.preheader:
                                                                                                                           ... undef, < 8 \times 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 ]
                                                                                                                                %29 = add nuw nsw <8 x i64> %vec.ind, %broadcast.splat
                                                                                                                                %30 = trunc < 8 \times i64 > %29 \text{ to } < 8 \times i32 > 
                                                                                                                                %31 = icmp sgt <8 x i32> %broadcast.splat27, %30
                                                                                                                                %32 = \text{extractelement} < 8 \times i32 > %30, i32 0
                                                                                                                                %33 = add nsw i32 %mul.i, %32
                                                                                                                                %34 = \text{sext i} 32 \% 33 \text{ to i} 64
                                                                                                                                %35 = getelementptr inbounds float, float* %2, i64 %34
                                                                                                                                %36 = bitcast float* %35 to <8 x float>*
                                                                                                                                %wide.masked.load = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x
                                                                                                                               ... float>* %36, i32 4, <8 x i1> %31, <8 x float> undef), !tbaa !12, !alias.scope
                                                                                                                                ... !16, !noalias !19
                                                                                                                                %37 = add nsw i32 %mul5.i, %32
                                                                                                                                %38 = \text{sext i} 32 \% 37 \text{ to i} 64
                                                                                                                               %39 = getelementptr inbounds float, float* %2, i64 %38 %40 = bitcast float* %39 to <8 x float>*
                                                                                                                                %wide.masked.load28 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x
                                                                                                                               ... float>* %40, i32 4, <8 x i1> %31, <8 x float> undef), !tbaa !12, !alias.scope
                                                                                                                                %41 = getelementptr inbounds float, float* %0, i64 %38
                                                                                                                                %42 = bitcast float* %41 to <8 x float>*
                                                                                                                                %wide.masked.load29 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x
                                                                                                                               ... float>* %42, i32 4, <8 x i1> %31, <8 x float> undef), !tbaa !12, !alias.scope
                                                                                                                                ... !24
                                                                                                                                %43 = fneg < 8 x float > %wide.masked.load28
                                                                                                                                %44 = call <8 x float> @llvm.fmuladd.v8f32(<8 x float> %43, <8 x float>
                                                                                                                                ... %wide.masked.load29, <8 x float> %wide.masked.load)
                                                                                                                                %45 = getelementptr inbounds float, float* %1, i64 %34
                                                                                                                                %46 = bitcast float* %45 to <8 x float>*
                                                                                                                                %wide.masked.load30 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x
                                                                                                                               ... float>* %46, i32 4, <8 x i1> %31, <8 x float> undef), !tbaa !12, !alias.scope
                                                                                                                                %47 = fdiv <8 x float> %44, %wide.masked.load30, !fpmath !26
                                                                                                                                %48 = bitcast float* %35 to <8 x float>*
                                                                                                                               call void @llvm.masked.store.v8f32.p0v8f32(<8 x float> %47, <8 x float>*
                                                                                                                                ... %48, i32 4, <8 x i1> %31), !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 ], [ %83,
                                                                                                                                %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>
                                        %70 = add nuw nsw i64 %_local_id_x.0, %9
                                                                                                                                %49 = add nuw nsw <8 x i64> %vec.ind.next, %broadcast.splat
                                        %conv.i = trunc i64 %70 to i32
                                                                                                                                %50 = \text{trunc} < 8 \times i64 > %49 \text{ to} < 8 \times i32 >
                                        %cmp.i = icmp slt i32 %conv.i, %4
                                                                                                                                %51 = icmp sgt <8 x i32> %broadcast.splat27, %50
                                        br il %cmp.i, label %if.then.i, label %if.end.r exit.i
                                                                                                                                %52 = \text{extractelement} < 8 \times i32 > %50, i32 0
                                                                                                     F
                                                                                                                                %53 = add nsw i32 %mul.i, %52
                                                                                                                                \%54 = \text{sext i} 32 \%53 \text{ to i} 64
                                                                                                                                %55 = getelementptr inbounds float, float* %2, i64 %54
                                                                                                                                \%56 = bitcast float* \%55 to < 8 x float>*
                                                                                                                                %wide.masked.load.1 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x
                                                                                                                               ... float>* %56, i32 4, <8 x i1> %51, <8 x float> undef), !tbaa !12, !alias.scope
                                                                                                                                ... !16, !noalias !19
                                                                                                                                %57 = add nsw i32 %mul5.i, %52
                                                                                                                                %58 = \text{sext i} 32 \% 57 \text{ to i} 64
                                                                                                                               %59 = getelementptr inbounds float, float* %2, i64 %58 %60 = bitcast float* %59 to <8 x float>*
                                                                                                                               %wide.masked.load28.1 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 ... x float>* %60, i32 4, <8 x i1> %51, <8 x float> undef), !tbaa !12,
                                                                                                                                ... !alias.scope !23
                                                                                                                               %61 = getelementptr inbounds float, float* %0, i64 %58 %62 = bitcast float* %61 to <8 x float>*
                                                                                                                                %wide.masked.load29.1 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8
                                                                                                                               ... x float>* %62, i32 4, <8 x i1> %51, <8 x float> undef), !tbaa !12,
                                                                                                                                ... !alias.scope !24
                                                                                                                                \%63 = \text{fneg} < 8 \text{ x float} > \% \text{wide.masked.load} 28.1
                                                                                                                                %64 = call <8 x float> @llvm.fmuladd.v8f32(<8 x float> %63, <8 x float>
                                                                                                                                ... %wide.masked.load29.1, <8 x float> %wide.masked.load.1)
                                                                                                                                %65 = getelementptr inbounds float, float* %1, i64 %54
                                                                                                                                %66 = bitcast float* %65 to <8 x float>*
                                                                                                                                %wide.masked.load30.1 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8
                                                                                                                               ... x float>* %66, i32 4, <8 x i1> %51, <8 x float> undef), !tbaa !12,
                                                                                                                               ... !alias.scope !25
                                                                                                                               %67 = fdiv <8 x float> %64, %wide.masked.load30.1, !fpmath !26 %68 = bitcast float* %55 to <8 x float>*
                                                                                                                               call void @llvm.masked.store.v8f32.p0v8f32(<8 x float> %67, <8 x float>*
                                                                                                                                ... %68, i32 4, <8 x i1> %51), !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>
                                                                                                                                %69 = icmp eq i64 %index.next.1, 256
                                                                                                                               br i1 %69, 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 %71 = 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
              %72 = load float, float* %arrayidx8.i, align 4, !tbaa !12
%arrayidx14.i = getelementptr inbounds float, float* %0, i64 %idxprom7.i
              %73 = load float, float* %arrayidx14.i, align 4, !tbaa !12
              %neg.i = fneg float %72
              %74 = tail call float @llvm.fmuladd.f32(float %neg.i, float %73, float %71)
              %arrayidx21.i = getelementptr inbounds float, float* %1, i64 %idxprom.i %75 = load float, float* %arrayidx21.i, align 4, !tbaa !12
              %div.i = fdiv float %74, %75, !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:
                             \%76 = \text{ or } i64 \% \text{ local id } x.0, 1
                             %77 = add nuw nsw i64 \ \%76, \%9
                                                                                                                adi_kernel6.exit.loopexit32:
                             %conv.i.1 = trunc i64 %77 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 %78 = 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
%79 = load float, float* %arrayidx8.i.1, align 4, !tbaa !12
%arrayidx14.i.1 = getelementptr inbounds float, float* %0, i64 %idxprom7.i.1
%80 = load float, float* %arrayidx14.i.1, align 4, !tbaa !12
%neg.i.1 = fneg float %79
%81 = tail call float @llvm.fmuladd.f32(float %neg.i.1, float %80, float
... %78) #5
%arrayidx21.i.1 = getelementptr inbounds float, float* %1, i64 %idxprom.i.1 %82 = load float, float* %arrayidx21.i.1, align 4, !tbaa !12
%div.i.1 = fdiv float %81, %82, !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:
                                                         \%83 = add nuw nsw i64 \% local id x.0, 2
                                                         %exitcond.1 = icmp eq i64 %83, 256
br i1 %exitcond.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
```