```
%9:
 %10 = sext i32 %4 to i64
 %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, !llvm.access.group !12
 %mul.i = mul nsw i32 %sub2.i, %4, !llvm.access.group !12
 %sub3.i = sub i32 -3, %3
 %sub4.i = add i32 %sub3.i, %4, !llvm.access.group !12
 %mul5.i = mul nsw i32 %sub4.i, %4, !llvm.access.group !12
 %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
```

```
vector.scevcheck:
 %14 = add nsw i64 %umax, -1
 %15 = mul i32 %sub2.i, %4
 %16 = trunc i64 %6 to i32
 %17 = shl i32 %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 = or i1 %21, %22
 %24 = mul i32 %sub4.i, %4
 %25 = add nsw i32 %24, %17
 %26 = trunc i64 %14 to i32
 %27 = add i32 %25, %26
 %28 = icmp slt i32 %27, %25
 %29 = icmp ugt i64 %14, 4294967295
 %30 = or i1 %28, %29
 %31 = or i1 %23, %30
 br i1 %31, label %pregion_for_entry.entry.i.preheader, label %vector.ph
```

```
vector.ph:
 %n.vec = and i64 %umax, -8
 br label %vector.body
```

```
vector.body:
 %index = phi i64 [ 0, %vector.ph ], [ %index.next, %vector.body ]
 %32 = add i64 %index, %mul.i.i, !llvm.access.group !12
 %33 = trunc i64 %32 to i32, !llvm.access.group !12
 %34 = add nsw i32 %mul.i, %33, !llvm.access.group !12
 %35 = sext i32 %34 to i64, !llvm.access.group !12
 %36 = getelementptr inbounds float, float* %2, i64 %35, !llvm.access.group
... !12
 %37 = bitcast float* %36 to <8 x float>*
 %wide.load = load <8 x float>, <8 x float>* %37, align 4, !tbaa !14,
... !llvm.access.group !12
 %38 = add nsw i32 %mul5.i, %33, !llvm.access.group !12
 %39 = sext i32 %38 to i64, !llvm.access.group !12
 %40 = getelementptr inbounds float, float* %2, i64 %39, !llvm.access.group
... !12
 %41 = bitcast float* %40 to <8 x float>*
 %wide.load4 = load <8 x float>, <8 x float>* %41, align 4, !tbaa !14,
... !llvm.access.group !12
 %42 = getelementptr inbounds float, float* %0, i64 %39, !llvm.access.group
... !12
 %43 = bitcast float* %42 to <8 x float>*
 %wide.load5 = load <8 x float>, <8 x float>* %43, align 4, !tbaa !14,
... !llvm.access.group !12
 %44 = fneg <8 x float> %wide.load4, !llvm.access.group !12
 %45 = call <8 x float> @llvm.fmuladd.v8f32(<8 x float> %44, <8 x float>
... %wide.load5, <8 x float> %wide.load), !llvm.access.group !12
 %46 = getelementptr inbounds float, float* %1, i64 %35, !llvm.access.group
... !12
 %47 = bitcast float* %46 to <8 x float>*
 %wide.load6 = load <8 x float>, <8 x float>* %47, align 4, !tbaa !14,
... !llvm.access.group !12
 %48 = fdiv <8 x float> %45, %wide.load6, !fpmath !18, !llvm.access.group !12
 %49 = bitcast float* %36 to <8 x float>*
 store <8 x float> %48, <8 x float>* %49, align 4, !tbaa !14,
... !llvm.access.group !12
 %index.next = add i64 %index, 8
 %50 = icmp eq i64 %index.next, %n.vec
 br i1 %50, 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_kernel6.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, %9 ], [ %n.vec,
... %middle.block ]
 br label %pregion_for_entry.entry.i
```

```
pregion_for_entry.entry.i:
 %_local_id_x.0 = phi i64 [ %56, %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, !llvm.access.group !12
 %conv.i = trunc i64 %add1.i.i to i32, !llvm.access.group !12
 %add.i = add nsw i32 %mul.i, %conv.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
 %51 = load float, float* %arrayidx.i, align 4, !tbaa !14, !llvm.access.group
... !12
 %add6.i = add nsw i32 %mul5.i, %conv.i, !llvm.access.group !12
 %idxprom7.i = sext i32 %add6.i to i64, !llvm.access.group !12
 %arrayidx8.i = getelementptr inbounds float, float* %2, i64 %idxprom7.i,
... !llvm.access.group !12
 %52 = load float, float* %arrayidx8.i, align 4, !tbaa !14,
... !llvm.access.group !12
 %arrayidx14.i = getelementptr inbounds float, float* %0, i64 %idxprom7.i,
... !llvm.access.group !12
 %53 = load float, float* %arrayidx14.i, align 4, !tbaa !14,
... !llvm.access.group !12
 %neg.i = fneg float %52, !llvm.access.group !12
 %54 = tail call float @llvm.fmuladd.f32(float %neg.i, float %53, float %51)
... #3, !llvm.access.group !12
 %arrayidx21.i = getelementptr inbounds float, float* %1, i64 %idxprom.i,
... !llvm.access.group !12
 %55 = load float, float* %arrayidx21.i, align 4, !tbaa !14,
... !llvm.access.group !12
 %div.i = fdiv float %54, %55, !fpmath !18, !llvm.access.group !12
 store float %div.i, float* %arrayidx.i, align 4, !tbaa !14,
... !llvm.access.group !12
 %56 = add nuw i64 %_local_id_x.0, 1
 %exitcond.not = icmp eq i64 %56, %umax
 br i1 %exitcond.not, label %adi_kernel6.exit.loopexit, label
... %pregion_for_entry.entry.i, !llvm.loop !22
```

```
adi_kernel6.exit.loopexit:
 br label %adi_kernel6.exit
```

```
adi_kernel6.exit:
 ret void
```

CFG for '_pocl_kernel_adi_kernel6' function