```
%11:
 %12 = sext i32 %2 to i64
 %13 = icmp slt i64 %12, 32
 %14 = select i1 %13, i64 %12, i64 32
 %15 = sext i32 %1 to i64
 %16 = icmp slt i64 %15, 8
 %17 = select i1 %16, i64 %15, i64 8
 %mul.i.i = shl i64 %8, 5
 %mul3.i.i = shl i64 %9, 3
 %mul6.i = mul i32 %6, %1
 %18 = icmp ugt i64 %14, 1
 %umax = select i1 %18, i64 %14, i64 1
 %19 = icmp ugt i64 %17, 1
 %umax1 = select i1 %19, i64 %17, i64 1
 %20 = add nsw i64 %umax, -1
 %21 = trunc i64 %9 to i32
 %22 = shl i32 %21, 3
 %23 = add i32 %mul6.i, %22
 %24 = mul i32 %23, %2
 %25 = trunc i64 %8 to i32
 %26 = shl i32 %25, 5
 %27 = add i32 %24, %26
 %scevgep4 = getelementptr float, float* %3, i64 %umax
 %scevgep9 = getelementptr float, float* %5, i64 %umax
 br label %pregion_for_entry.pregion_for_init.i
```

```
pregion_for_entry.pregion_for_init.i:
 %_local_id_y.0 = phi i64 [ 0, %11 ], [ %59, %pregion_for_end.i ]
 %28 = trunc i64 %_local_id_y.0 to i32
 %29 = mul i32 %28, %2
 %30 = add i32 %29, %27
 %add6.i.i = add i64 %_local_id_y.0, %mul3.i.i
 %conv2.i = trunc i64 %add6.i.i to i32
 %reass.add.i = add i32 %mul6.i, %conv2.i
 %reass.mul.i = mul i32 %reass.add.i, %2
 %min.iters.check = icmp ult i64 %umax, 32
 br i1 %min.iters.check, label %pregion_for_entry.entry.i.preheader, label
... %vector.scevcheck
```

```
vector.scevcheck:
 %31 = trunc i64 %20 to i32
 %32 = add i32 %30, %31
 %33 = icmp slt i32 %32, %30
 %34 = icmp ugt i64 %20, 4294967295
 %35 = or i1 %33, %34
 br i1 %35, label %pregion_for_entry.entry.i.preheader, label %vector.memcheck
```

```
vector.memcheck:
 %bound0 = icmp ugt float* %scevgep9, %3
 %bound1 = icmp ugt float* %scevgep4, %5
 %found.conflict = and i1 %bound0, %bound1
 br i1 %found.conflict, label %pregion_for_entry.entry.i.preheader, label
... %vector.ph
```

```
vector.ph:
 %n.vec = and i64 %umax, -32
 br label %vector.body
```

```
vector.body:
 %index = phi i64 [ 0, %vector.ph ], [ %index.next, %vector.body ]
 %36 = add i64 %index, %mul.i.i
 %37 = trunc i64 %36 to i32
 %38 = add i32 %reass.mul.i, %37
 %39 = sext i32 %38 to i64
 %40 = getelementptr inbounds float, float* %5, i64 %39
 %41 = bitcast float* %40 to <8 x float>*
 %wide.load = load <8 x float>, <8 x float>* %41, align 4, !tbaa !12,
... !alias.scope !16
 %42 = getelementptr inbounds float, float* %40, i64 8
 %43 = bitcast float* %42 to <8 x float>*
 %wide.load12 = load <8 x float>, <8 x float>* %43, align 4, !tbaa !12,
... !alias.scope !16
 %44 = getelementptr inbounds float, float* %40, i64 16
 %45 = bitcast float* %44 to <8 x float>*
 %wide.load13 = load <8 x float>, <8 x float>* %45, align 4, !tbaa !12,
... !alias.scope !16
 %46 = getelementptr inbounds float, float* %40, i64 24
 %47 = bitcast float* %46 to <8 x float>*
 %wide.load14 = load <8 x float>, <8 x float>* %47, align 4, !tbaa !12,
... !alias.scope !16
 %48 = getelementptr inbounds float, float* %3, i64 %39
 %49 = bitcast float* %48 to <8 x float>*
 store <8 x float> %wide.load, <8 x float>* %49, align 4, !tbaa !12,
... !alias.scope !19, !noalias !16, !llvm.access.group !21
 %50 = getelementptr inbounds float, float* %48, i64 8
 %51 = bitcast float* %50 to <8 x float>*
 store <8 x float> %wide.load12, <8 x float>* %51, align 4, !tbaa !12,
... !alias.scope !19, !noalias !16, !llvm.access.group !21
 %52 = getelementptr inbounds float, float* %48, i64 16
 %53 = bitcast float* %52 to <8 x float>*
 store <8 x float> %wide.load13, <8 x float>* %53, align 4, !tbaa !12,
... !alias.scope !19, !noalias !16, !llvm.access.group !21
 %54 = getelementptr inbounds float, float* %48, i64 24
 %55 = bitcast float* %54 to <8 x float>*
 store <8 x float> %wide.load14, <8 x float>* %55, align 4, !tbaa !12,
... !alias.scope !19, !noalias !16, !llvm.access.group !21
 %index.next = add i64 %index, 32
 %56 = icmp eq i64 %index.next, %n.vec
 br i1 %56, label %middle.block, label %vector.body, !llvm.loop !24
```

```
middle.block:
 %cmp.n = icmp eq i64 %umax, %n.vec
 br i1 %cmp.n, label %pregion_for_end.i, 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, %pregion_for_entry.pregion_for_init.i ], [ %n.vec, %middle.block ]
 br label %pregion_for_entry.entry.i
```

```
pregion_for_entry.entry.i:
 %_local_id_x.0 = phi i64 [ %58, %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
 %add8.i = add i32 %reass.mul.i, %conv.i
 %idxprom.i = sext i32 %add8.i to i64
 %arrayidx.i = getelementptr inbounds float, float* %5, i64 %idxprom.i
 %57 = load float, float* %arrayidx.i, align 4, !tbaa !12
 %arrayidx15.i = getelementptr inbounds float, float* %3, i64 %idxprom.i
 store float %57, float* %arrayidx15.i, align 4, !tbaa !12,
... !llvm.access.group !21
 %58 = add nuw i64 %_local_id_x.0, 1
 %exitcond.not = icmp eq i64 %58, %umax
 br i1 %exitcond.not, label %pregion_for_end.i.loopexit, label
... %pregion_for_entry.entry.i, !llvm.loop !27
```

```
pregion_for_end.i.loopexit:
 br label %pregion_for_end.i
```

```
pregion_for_end.i:
 %59 = add nuw i64 %_local_id_y.0, 1
 %exitcond2.not = icmp eq i64 %59, %umax1
 br i1 %exitcond2.not, label %doitgen_kernel2.exit, label
... %pregion_for_entry.pregion_for_init.i, !llvm.loop !28
```

```
doitgen_kernel2.exit:
 ret void
```

CFG for '_pocl_kernel_doitgen_kernel2' function