```
%11 = \text{sext i} 32 \% 5 \text{ to i} 64
                                                                                                                                                                                                      %12 = icmp slt i64 %11, 32
                                                                                                                                                                                                       %13 = select i1 %12, i64 %11, i64 32
                                                                                                                                                                                                       %14 = icmp slt i64 %11, 8
                                                                                                                                                                                                       %15 = select i1 %14, i64 %11, i64 8
                                                                                                                                                                                                        %mul.i.i = shl i64 %7, 5
                                                                                                                                                                                                       %mul3.i.i = shl i64 %8, 3
                                                                                                                                                                                                       %cmp742.i = icmp sgt i32 %4, 0, !llvm.access.group !12 %wide.trip.count.i = zext i32 %4 to i64
                                                                                                                                                                                                        %16 = icmp ugt i64 \%13, 1
                                                                                                                                                                                                       %umax = select i1 %16, i64 %13, i64 1
                                                                                                                                                                                                       %17 = icmp ugt i64 \%15, 1
                                                                                                                                                                                                       %umax3 = select i1 %17, i64 %15, i64 1
                                                                                                                                                                                                       %min.iters.check = icmp ult i64 %umax3, 8
                                                                                                                                                                                                       br i1 %min.iters.check, label
                                                                                                                                                                                                       ... %pregion for entry.pregion for init.i.preheader, label %vector.ph
                                                                                                                                                                                                                                           Τ
                                                                                                                                                                                                                                                                                                                                 F
                                                                                                                                                                                                                                                         vector.ph:
                                                                                                                                                                                                                                                           %n.vec = and i64 %umax3, -8
                                                                                                                                                                                                                                                           %broadcast.splatinsert = insertelement <8 x i64> undef, i64 %mul3.i.i, i32 0
                                                                                                                                                                                                                                                           %broadcast.splat = shufflevector <8 x i64> %broadcast.splatinsert, <8 x i64>
                                                                                                                                                                                                                                                            .. undef, <8 x i32> zeroinitializer
                                                                                                                                                                                                                                                           %broadcast.splatinsert5 = insertelement <8 x i32> undef, i32 %5, i32 0
                                                                                                                                                                                                                                                           %broadcast.splat6 = shufflevector <8 x i32> %broadcast.splatinsert5, <8 x
                                                                                                                                                                                                                                                           ... i32> undef, <8 x i32> zeroinitializer
                                                                                                                                                                                                                                                           %broadcast.splatinsert7 = insertelement <8 x i32> undef, i32 %4, i32 0
                                                                                                                                                                                                                                                           %broadcast.splat8 = shufflevector <8 x i32> %broadcast.splatinsert7, <8 x
                                                                                                                                                                                                                                                           ... i32> undef, <8 x i32> zeroinitializer
                                                                                                                                                                                                                                                           %broadcast.splatinsert10 = insertelement <8 x i64> undef, i64 %mul.i.i, i32 0
                                                                                                                                                                                                                                                          %broadcast.splat11 = shufflevector <8 x i64> %broadcast.splatinsert10, <8 x
                                                                                                                                                                                                                                                            .. i64> undef, <8 x i32> zeroinitializer
                                                                                                                                                                                                                                                           %broadcast.splatinsert12 = insertelement <8 x float> undef, float %3, i32 0
                                                                                                                                                                                                                                                           %broadcast.splat13 = shufflevector <8 x float> %broadcast.splatinsert12, <8
                                                                                                                                                                                                                                                            .. x float> undef, <8 x i32> zeroinitializer
                                                                                                                                                                                                                                                           %broadcast.splatinsert21 = insertelement <8 x float> undef, float %2, i32 0
                                                                                                                                                                                                                                                           %broadcast.splat22 = shufflevector <8 x float> %broadcast.splatinsert21, <8
                                                                                                                                                                                                                                                           ... x float> undef, <8 x i32> zeroinitializer
                                                                                                                                                                                                                                                           %broadcast.splatinsert24 = insertelement <8 x i64> undef, i64
                                                                                                                                                                                                                                                           ... %wide.trip.count.i, i32 0
                                                                                                                                                                                                                                                           %broadcast.splat25 = shufflevector <8 x i64> %broadcast.splatinsert24, <8 x
                                                                                                                                                                                                                                                           ... i64> undef, <8 x i32> zeroinitializer
                                                                                                                                                                                                                                                           %broadcast.splatinsert28 = insertelement <8 x i64> undef, i64 %umax, i32 0
                                                                                                                                                                                                                                                           %broadcast.splat29 = shufflevector <8 x i64> %broadcast.splatinsert28, <8 x
                                                                                                                                                                                                                                                            .. i64> undef, <8 x i32> zeroinitializer
                                                                                                                                                                                                                                                           br label %vector.body
                                                                                                                                                                                                                                                                                      vector.body:
                                                                                                                                                                                                                                                                                      %index = phi i64 [ 0, %vector.ph ], [ %index.next, %pregion for end.i30 ]
                                                                                                                                                                                                                                                                                       %vec.ind = phi < 8 \times 164 > [< 164 \ 0, 164 \ 1, 164 \ 2, 164 \ 3, 164 \ 4, 164 \ 5, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6
                                                                                                                                                                                                                                                                                      ... i64 7>, %vector.ph ], [ %vec.ind.next, %pregion_for_end.i30 ]
                                                                                                                                                                                                                                                                                      %18 = add <8 x i64> %vec.ind, %broadcast.splat, !llvm.access.group !12
                                                                                                                                                                                                                                                                                       %19 = trunc <8 x i64> %18 to <8 x i32>, !llvm.access.group !12
                                                                                                                                                                                                                                                                                      %20 = mul nsw <8 x i32> %broadcast.splat6, %19, !llvm.access.group !12 %21 = mul nsw <8 x i32> %broadcast.splat8, %19
                                                                                                                                                                                                                                                                                      %22 = \text{sext} < 8 \times i32 > \%21 \text{ to } < 8 \times i64 > \%21
                                                                                                                                                                                                                                                                                      br label %pregion for entry.entry.i9
                                                                                                                                                                                                                                                                              pregion for entry.entry.i9:
                                                                                                                                                                                                                                                                               % \sqrt{\text{ec.phi}} = \text{phi } < 8 \times 164 > [\text{zeroinitializer}, \text{%vector.body}], [\text{%41}, ]
                                                                                                                                                                                                                                                                                ... %if.end.r exit.i27 1
                                                                                                                                                                                                                                                                               %23 = add <8 x i64> %vec.phi, %broadcast.splat11, !llvm.access.group !12 %24 = trunc <8 x i64> %23 to <8 x i32>, !llvm.access.group !12
                                                                                                                                                                                                                                                                                %25 = add nsw <8 x i32> %20, %24, !llvm.access.group !12
                                                                                                                                                                                                                                                                               \%26 = \text{sext} < 8 \times i32 > \%25 \text{ to } < 8 \times i64 >, !llvm.access.group !12
                                                                                                                                                                                                                                                                               %27 = getelementptr inbounds float, float* %1, <8 x i64> %26,
                                                                                                                                                                                                                                                                                ...!llvm.access.group!12
                                                                                                                                                                                                                                                                               %wide.masked.gather = call <8 x float> @llvm.masked.gather.v8f32.v8p0f32(<8 ... x float*> %27, i32 4, <8 x i1> <i1 true, i1 true,
                                                                                                                                                                                                                                                                               ... i1 true, i1 true, i1 true, <8 x float> undef), !tbaa !15, !llvm.access.group
                                                                                                                                                                                                                                                                              ... !12
                                                                                                                                                                                                                                                                                %28 = fmul <8 x float> %wide.masked.gather, %broadcast.splat13,
                                                                                                                                                                                                                                                                                ...!llvm.access.group!12
                                                                                                                                                                                                                                                                               call void @llvm.masked.scatter.v8f32.v8p0f32(<8 x float> %28, <8 x float*> ... %27, i32 4, <8 x i1> <i1 true, i1 true, i
                                                                                                                                                                                                                                                                               ... i1 true, i1 true>), !tbaa !15, !llvm.access.group !12
                                                                                                                                                                                                                                                                               br i1 %cmp742.i, label %for.body.lr.ph.i16, label %if.end.r exit.i27
                                                                                                                                                                                                                           for.body.lr.ph.i16:
                                                                                                                                                                                                                             %29 = mul nsw <8 x i32> %broadcast.splat8, %24, !llvm.access.group !12
                                                                                                                                                                                                                             %30 = \text{sext} < 8 \times i32 > %29 \text{ to } < 8 \times i64 >, !llvm.access.group !12
                                                                                                                                                                                                                             br label %for.body.i17
                                                                                                                                                                                                           for.body.i17:
                                                                                                                                                                                                            %vec.phi18 = phi < 8 \times i64 > [ %38, %for.body.i17 ], [ zeroinitializer, ]
                                                                                                                                                                                                            ... %for.body.lr.ph.i16 ]
                                                                                                                                                                                                            %vec.phi19 = phi <8 x float> [ \%37, \%for.body.i17 ], [ \%28,
                                                                                                                                                                                                            ... %for.body.lr.ph.i16 ]
                                                                                                                                                                                                            %31 = add nsw <8 x i64> %vec.phi18, %22, !llvm.access.group !12
                                                                                                                                                                                                            %32 = getelementptr inbounds float, float* %0, <8 x i64> %31,
                                                                                                                                                                                                            ...!llvm.access.group!12
                                                                                                                                                                                                           %wide.masked.gather20 = call <8 x float> ... @llvm.masked.gather.v8f32.v8p0f32(<8 x float*> %32, i32 4, <8 x i1> <i1 true,
                                                                                                                                                                                                           ... i1 true, i7 true, i1 true,
                                                                                                                                                                                                            ... undef), !tbaa !15, !llvm.access.group !12
                                                                                                                                                                                                            %33 = fmul <8 x float> %wide.masked.gather20, %broadcast.splat22,
                                                                                                                                                                                                            ...!llvm.access.group!12
                                                                                                                                                                                                            %34 = add nsw <8 x i64> %vec.phi18, %30, !llvm.access.group !12
                                                                                                                                                                                                            %35 = getelementptr inbounds float, float* %0, <8 x i64> %34,
                                                                                                                                                                                                            ...!llvm.access.group!12
                                                                                                                                                                                                            %wide.masked.gather23 = call <8 x float>
                                                                                                                                                                                                           ... @llvm.masked.gather.v8f32.v8p0f32(<8 x float*> %35, i32 4, <8 x i1> <i1 true, i1 true, i1
                                                                                                                                                                                                            ... undef), !tbaa !15, !llvm.access.group !12
                                                                                                                                                                                                             %36 = fmul <8 x float> %33, %wide.masked.gather23, !llvm.access.group !12
                                                                                                                                                                                                            %37 = fadd <8 x float> %vec.phi19, %36, !llvm.access.group !12
                                                                                                                                                                                                            call void @llvm.masked.scatter.v8f32.v8p0f32(<8 x float> %37, <8 x float*>
                                                                                                                                                                                                           ... %27, i32 4, <8 x i1> <i1 true, i1 t
                                                                                                                                                                                                            ... i64 1, i64 1>, !llvm.access.group !12

%39 = icmp eq <8 x i64> %38, %broadcast.splat25, !llvm.access.group !12
                                                                                                                                                                                                            %40 = \text{extractelement} < 8 \times i1 > %39, i32 0
                                                                                                                                                                                                            br i1 %40, label %if.end.r exit.i27.loopexit, label %for.body.i17
                                                                                                                                                                                                                                                                                          if.end.r exit.i27.loopexit:
                                                                                                                                                                                                                                                                                           br label %if.end.r exit.i27
                                                                                                                                                                                                                                                                                           if.end.r exit.i27:
                                                                                                                                                                                                                                                                                           %41 = add nuw <8 x i64> %vec.phi, <i64 1, i64 1, i64 1, i64 1, i64 1, i64 1,
                                                                                                                                                                                                                                                                                            ... i64 1, i64 1>
                                                                                                                                                                                                                                                                                            %42 = icmp eq <8 x i64> %41, %broadcast.splat29
                                                                                                                                                                                                                                                                                            %43 = \text{extractelement} < 8 \times i1 > %42, i32 0
                                                                                                                                                                                                                                                                                            br i1 %43, label %pregion for end.i30, label %pregion for entry.entry.i9
                                                                                                                                                                                                                                                                pregion for end.i30:
                                                                                                                                                                                                                                                                 %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>
                                                                                                                                                                                                                                                                 %44 = icmp eq i64 %index.next, %n.vec
                                                                                                                                                                                                                                                                br i1 %44, label %middle.block, label %vector.body, !llvm.loop !19
                                                                                                                                                                                                                                                                                                                                                                                                        F
                                                                                                                                                                                                                                              middle.block:
                                                                                                                                                                                                                                                %cmp.n = icmp eq i64 %umax3, %n.vec
                                                                                                                                                                                                                                                br i1 %cmp.n, label %syrk kernel.exit, label
                                                                                                                                                                                                                                                ... %pregion for entry.pregion for init.i.preheader
                                                                                                                pregion_for_entry.pregion_for_init.i.preheader:
    %_local_id_y.0.ph = phi i64 [ 0, %10 ], [ %n.vec, %middle.block ]
                                                                                                                 br label %pregion for entry.pregion for init.i
                                                                                                 pregion_for_entry.pregion_for_init.i:
                                                                                                  %_local_id_y.0 = phi i64 [ %56, %pregion_for_end.i ], [ %_local_id_y.0.ph, ... %pregion_for_entry.pregion_for_init.i.preheader ] %add6.i.i = add i64 %_local_id_y.0, %mul3.i.i, !llvm.access.group !12
                                                                                                   %conv2.i = trunc i64 %add6.i.i to i32, !llvm.access.group !12
                                                                                                   %mul.i = mul nsw i32 %conv2.i, %5, !llvm.access.group !12
                                                                                                   %mul9.i = mul nsw i32 %conv2.i, %4
                                                                                                   %45 = sext i32 %mul9.i to i64
                                                                                                   br label %pregion for entry.entry.i
                                                               pregion for entry.entry.i:
                                                                % local id \bar{x}.0 = phi i64 [ 0, %pregion for entry.pregion for init.i ], [
                                                                ... \( \bar{8} 55, \bar{8} if.end.r \) exit.i ]
                                                               %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* %1, i64 %idxprom.i,
                                                                 ...!llvm.access.group!12
                                                                %46 = load float, float* %arrayidx.i, align 4, !tbaa !15, !llvm.access.group
                                                                ... !12
                                                               %mul6.i = fmul float %46, %3, !llvm.access.group !12 store float %mul6.i, float* %arrayidx.i, align 4, !tbaa !15,
                                                                 ...!llvm.access.group!12
                                                                br i1 %cmp742.i, label %for.body.lr.ph.i, label %if.end.r_exit.i,
                                                                ...!llvm.access.group!12
                           for.body.lr.ph.i:
                            %mul14.i = mul nsw i32 %conv.i, %4, !llvm.access.group !12
                            %47 = sext i32 %mul14.i to i64, !llvm.access.group !12
                            br label %for.body.i, !llvm.access.group !12
for.body.i:
 %indvars.iv.next.i2 = phi i64 [ %indvars.iv.next.i, %for.body.i ], [ 0,
  .. %for.body.lr.ph.i ]
 %48 = phi float [ %54, %for.body.i ], [ %mul6.i, %for.body.lr.ph.i ]
 %49 = add nsw i64 %indvars.iv.next.i2, %45, !llvm.access.group !12
 %arrayidx12.i = getelementptr inbounds float, float* %0, i64 %49,
 ...!llvm.access.group!12
%50 = load float, float* %arrayidx12.i, align 4, !tbaa!15,
 ...!llvm.access.group!12
 %mul13.i = fmul float %50, %2, !llvm.access.group !12
 %51 = add nsw i64 %indvars.iv.next.i2, %47, !llvm.access.group !12
 %arrayidx17.i = getelementptr inbounds float, float* %0, i64 %51,
 ...!llvm.access.group!12
%52 = load float, float* %arrayidx17.i, align 4, !tbaa!15,
...!llvm.access.group!12
%53 = fmul float %mul13.i, %52,!llvm.access.group!12
 %54 = fadd float %48, %53, !llvm.access.group !12
 store float %54, float* %arrayidx.i, align 4, !tbaa !15, !llvm.access.group
  ...!12
 %indvars.iv.next.i = add nuw nsw i64 %indvars.iv.next.i2, 1,
  ..!llvm.access.group!12
 %exitcond.not.i = icmp eq i64 %indvars.iv.next.i, %wide.trip.count.i,
 ...!llvm.access.group!12
 br i1 %exitcond.not.i, label %if.end.r exit.i.loopexit, label %for.body.i,
 ...!llvm.loop!22,!llvm.access.group!12
                                                                                                                                      F
                                                                       if.end.r exit.i.loopexit:
                                                                       br label %if.end.r exit.i
                                                                                                            if.end.r exit.i:
                                                                                                             \%55 = add nuw i64 \% local id x.0, 1
                                                                                                             %exitcond.not = icmp eq i6\overline{4} %55, %umax
                                                                                                            br i1 %exitcond.not, label %pregion for end.i, label
                                                                                                            ... %pregion for entry.entry.i, !llvm.loop !24
                                                                                                                               pregion for end.i:
                                                                                                                                 \%56 = add nuw i64 % local id y.0, 1
                                                                                                                                %exitcond4.not = icm\bar{p} eq i\bar{6}4 %56, %umax3
                                                                                                                                br i1 %exitcond4.not, label %syrk kernel.exit.loopexit, label
                                                                                                                                 ... %pregion for entry pregion for init.i, !llvm.loop !27
                                                                                                                                                                           syrk kernel.exit.loopexit:
                                                                                                                                                                           br label %syrk kernel.exit
```

%10: