entry: %mem = getelementptr inbounds %struct.jpeg\_compress\_struct, ... %struct.jpeg\_compress\_struct\* %cinfo, i64 0, i32 1 %0 = load %struct.jpeg\_memory\_mgr\*, %struct.jpeg\_memory\_mgr\*\* %mem, align 8, ...!tbaa!3 %alloc\_small = getelementptr inbounds %struct.jpeg\_memory\_mgr, ... %struct.jpeg\_memory\_mgr\* %0, i64 0, i32 0 %1 = load i8\* (%struct.jpeg\_common\_struct\*, i32, i64)\*, i8\* ... (%struct.jpeg\_common\_struct\*, i32, i64)\*\* %alloc\_small, align 8, !tbaa !11 %2 = bitcast %struct.jpeg\_compress\_struct\* %cinfo to ... %struct.jpeg\_common\_struct\* %call = tail call i8\* %1(%struct.jpeg\_common\_struct\* %2, i32 1, i64 192) #3 %coef1 = getelementptr inbounds %struct.jpeg\_compress\_struct, ... %struct.jpeg\_compress\_struct\* %cinfo, i64 0, i32 54 %3 = bitcast %struct.jpeg\_c\_coef\_controller\*\* %coef1 to i8\*\* store i8\* %call, i8\*\* %3, align 8, !tbaa !14 %start\_pass = bitcast i8\* %call to void (%struct.jpeg\_compress\_struct\*, ... i32)\*\* store void (%struct.jpeg\_compress\_struct\*, i32)\* @start\_pass\_coef, void ... (%struct.jpeg\_compress\_struct\*, i32)\*\* %start\_pass, align 8, !tbaa !15 %tobool = icmp eq i32 %need\_full\_buffer, 0 br i1 %tobool, label %if.else, label %if.then, !prof!18 F for.body: %indvars.iv56 = phi i64 [ 0, %for.body.lr.ph ], [ %indvars.iv.next57, ... %for.body ] %compptr.054 = phi %struct.jpeg\_component\_info\* [ %5, %for.body.lr.ph ], [ ... %incdec.ptr, %for.body ] %7 = load %struct.jpeg\_memory\_mgr\*, %struct.jpeg\_memory\_mgr\*\* %mem, align 8, ...!tbaa!3 %request\_virt\_barray = getelementptr inbounds %struct.jpeg\_memory\_mgr, ... %struct.jpeg\_memory\_mgr\* %7, i64 0, i32 5 %8 = load %struct.jvirt\_barray\_control\* (%struct.jpeg\_common\_struct\*, i32, ... i32, i32, i32, i32)\*, %struct.jvirt\_barray\_control\* ... (%struct.jpeg\_common\_struct\*, i32, i32, i32, i32, i32)\*\* ... %request\_virt\_barray, align 8, !tbaa !22 %width\_in\_blocks = getelementptr inbounds %struct.jpeg\_component\_info, ... %struct.jpeg\_component\_info\* %compptr.054, i64 0, i32 7 %9 = load i32, i32\* %width\_in\_blocks, align 4, !tbaa !23 %conv = zext i32 %9 to i64%h\_samp\_factor = getelementptr inbounds %struct.jpeg\_component\_info, ... %struct.jpeg\_component\_info\* %compptr.054, i64 0, i32 2 %10 = load i32, i32\* %h\_samp\_factor, align 8, !tbaa !25 %conv3 = sext i32 %10 to i64 %call4 = tail call i64 @jround\_up(i64 %conv, i64 %conv3) #3 %conv5 = trunc i64 %call4 to i32 %height\_in\_blocks = getelementptr inbounds %struct.jpeg\_component\_info, .. %struct.jpeg\_component\_info\* %compptr.054, i64 0, i32 8 %11 = load i32, i32\* %height\_in\_blocks, align 8, !tbaa !26 %conv6 = zext i32 %11 to i64 %v\_samp\_factor = getelementptr inbounds %struct.jpeg\_component\_info, ... %struct.jpeg\_component\_info\* %compptr.054, i64 0, i32 3 %12 = load i32, i32\* %v\_samp\_factor, align 4, !tbaa !27 %conv7 = sext i32 %12 to i64 %call8 = tail call i64 @jround up(i64 %conv6, i64 %conv7) #3 %conv9 = trunc i64 %call8 to i32 %13 = load i32, i32\* %v\_samp\_factor, align 4, !tbaa !27 %call11 = tail call %struct.jvirt barray control\* ... %8(%struct.jpeg\_common\_struct\* nonnull %2, i32 1, i32 0, i32 %conv5, i32 ... %conv9, i32 %13) #3 %arrayidx = getelementptr inbounds [10 x %struct.jvirt\_barray\_control\*], [10 ... x %struct.jvirt\_barray\_control\*]\* %6, i64 0, i64 %indvars.iv56 store %struct.jvirt barray control\* %call11, %struct.jvirt barray control\*\* ... %arrayidx, align 8, !tbaa !28 %indvars.iv.next57 = add nuw nsw i64 %indvars.iv56, 1 %incdec.ptr = getelementptr inbounds %struct.jpeg component info, ... %struct.jpeg\_component\_info\* %compptr.054, i64 1 %14 = load i32, i32\* %num\_components, align 4, !tbaa !19 %15 = sext i 32 % 14 to i 64%cmp = icmp slt i64 %indvars.iv.next57, %15 br i1 %cmp, label %for.body, label %if.end.loopexit, !prof!20 F if.end: ret void