br i1 %cmp4, label %for.body5, label %for.end13 for.body5: ; preds = % for.cond3%4 = load i 32, i 32* % i 2, align 4%mul = mul i32 2, %4 %add = add i32 %mul. 1 %idxprom6 = zext i32 %add to i64 %arrayidx7 = getelementptr inbounds [100 x i32], [100 x i32]* %c, i64 0, i64 %idxprom6 %5 = load i32, i32* %arrayidx7, align 4 %6 = load i32, i32*%i2, align 4%mul8 = mul i32 3, %6 %sub = sub i32 %mul8, 5 %idxprom9 = zext i32 %sub to i64 %arrayidx10 = getelementptr inbounds [100 x i32], [100 x i32]* %c, i64 0, i64 %idxprom9 store i32 %5, i32* %arrayidx10, align 4 br label %for.inc11

%3 = load i32, i32* %i2, align 4

%cmp4 = icmp ult i32 %3, 100

; preds = %for.inc11, %for.end

for.cond3:

for.end: ; preds = %for.cond store i32 2, i32* %i2, align 4 br label %for.cond3

for.inc11: ; preds = %for.body5
%7 = load i32, i32* %i2, align 4
%inc12 = add i32 %7, 1
store i32 %inc12, i32* %i2, align 4
br label %for.cond3

for.inc:

for.cond:

; preds = %for.inc, %entry

%0 = load i32, i32* %i, align 4
%cmp = icmp ult i32 %0, 100
br i1 %cmp, label %for.body, label %for.end

%call = %coccall v

; preds = % for.body

%2 = load i 32, i 32* %i, align 4

%inc = add i32 %2. 1

store i32 %inc, i32* %i, align 4

br label %for.cond

%retval = alloca i32, align 4
%c = alloca [100 x i32], align 16
%i = alloca i32, align 4
%i2 = alloca i32, align 4
store i32 0, i32* %retval, align 4
%call = call i64 @time(i64* null) #2
%conv = trunc i64 %call to i32
call void @srand(i32 %conv) #2
store i32 0, i32* %i, align 4
br label %for.cond

for.end13: ; preds = %for.cond3 ret i32 0