; preds = %for.body | | | for.cond: %2 = load i32, i32* %i, align 4 %inc = add i32 %2, 1 store i32 %inc, i32* %i, align 4 br label %for.cond

; preds = %for.inc, %entry $\%0 = \text{load i} 32, \, \text{i} 32*\% i, \, \text{align 4}$ %cmp = icmp ult i32 %0, 100 br i1 %cmp, label %for.body, label %for.end

%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

; preds = % for.body5for.inc13: %9 = load i32, i32* %i2, align 4 %inc14 = add i32 %9, 1 store i32 %inc14, i32* %i2, align 4 br label %for.cond3

for.body: ; preds = %for.cond %call1 = call i32 @rand() #2 %rem = urem i32 %call1, 10 %1 = load i32, i32* %i, align 4 for.end15: %idxprom = zext i32 %1 to i64 %arrayidx = getelementptr inbounds [100 x i32], [100 x i32]* %c, i64 0, i64 %idxprom store i32 %rem, i32* %arrayidx, align 4 br label %for.inc

ret i32 0

; preds = %for.cond3 %4 = load i32, i32* %i2, align 4 for.body5: %sub = sub i32 %4, 1 %idxprom6 = zext i32 %sub 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 %sub8 = sub i32 %6, 2 ; preds = % for.cond3%idxprom9 = zext i32 %sub8 to i64 %arrayidx10 = getelementptr inbounds [100 x i32], [100 x i32]* %c, i64 0, i64 %idxprom9 %7 = load i32, i32* %arrayidx10, align 4 %add = add i32 %5, %7 %8 = load i32, i32* %i2, align 4 %idxprom11 = zext i32 %8 to i64 %arrayidx12 = getelementptr inbounds [100 x i32], [100 x i32]* %c, i64 0, i64 %idxprom11 store i32 %add, i32* %arrayidx12, align 4

br label %for.inc13

; preds = %for.cond %3 = load i32, i32* %i2, align 4 store i32 2, i32* %i2, align 4 %cmp4 = icmp ult i32 %3, 100 br label %for.cond3 br i1 %cmp4, label %for.body5, label %for.end15

; preds = %for.inc13, %for.end