

[6/11]

```
[14/15]
for.end: ; preds = %for.cond
store i32 0, i32* %i6, align 4
br label %for.cond7
```

; preds = % for.body

[9/10]

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

%inc = add i32 %3, 1

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

br label %for.cond

[0/1] for.body9: ; preds = % for cond7 %5 = load i32, i32* % i6, align 4%idxprom10 = zext i32 %5 to i64 %arrayidx11 = getelementptr inbounds [100 x i32], [100 x i32]* %a, i64 0, i64 %idxprom10 %6 = load i32, i32* %arrayidx11, align 4 %7 = load i32, i32* %i6, align 4 %idxprom12 = zext i32 %7 to i64 %arrayidx13 = getelementptr inbounds [100 x i32], [100 x i32]* %b, i64 0, i64 %idxprom12 $\sqrt[6]{8}$ = load i32, i32* %arrayidx13, align 4 %add = add i32 %6, %8 %9 = load i 32, i 32* % i 6, align 4% idxprom 14 = zext i 32 % 9 to i 64%arrayidx15 = getelementptr inbounds [100 x i32], [100 x i32]* %a, i64 0, i64 %idxprom14 store i32 %add, i32* %arrayidx15, align 4 br label %for.inc16

for.cond7:

[4/5]

%4 = load i32, i32* %i6, align 4

%cmp8 = icmp ult i32 %4, 100

br i1 %cmp8, label %for.body9, label %for.end18

; preds = %for.inc16, %for.end

for.inc16:

entry:
%retval = alloca i32, align 4
%a = alloca [100 x i32], align 16
%b = alloca [100 x i32], align 16
%i = alloca i32, align 4
%i6 = 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.body9

[2/3]

%10 = load i32, i32* %i6, align 4

%inc17 = add i32 %10, 1

store i32 %inc17, i32* %i6, align 4

br label %for.cond7

[12/13]

for.end18: [16/17]for.end18: ; preds = % for.cond7
ret i32 0