entry:
%retval = alloca i32, align 4
%a = alloca [100 x i32], align 16
%b = alloca [100 x i32], align 16
%c = alloca [100 x i32], align 16
%i = alloca i32, align 4
store i32 0, i32* %retval, align 4
br label %for.cond

for.cond:

[8/9]
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

```
; preds = %for.cond
               for.body:
                            %1 = load i32, i32* %i, align 4
                           %idxprom = zext i32 %1 to i64
 %arrayidx = getelementptr inbounds [100 x i32], [100 x i32]* %a, i64 0, i64 %idxprom
                       %2 = load i32, i32* %arrayidx, align 4
                           %3 = load i32, i32* \%i, align 4
                           %idxprom1 = zext i32 %3 to i64
%arrayidx2 = getelementptr inbounds [100 x i32], [100 x i32]* %b, i64 0, i64 %idxprom1
                       %4 = load i32, i32* %arrayidx2, align 4
                               %add = add i32 %2, %4
                           %5 = load i32, i32* %i, align 4
                           %idxprom3 = zext i32 %5 to i64
%arrayidx4 = getelementptr inbounds [100 x i32], [100 x i32]* %c, i64 0, i64 %idxprom3
                      store i32 %add, i32* %arrayidx4, align 4
                                  br label %for.inc
```

```
for.inc: ; preds = %for.body %6 = load i32, i32* %i, align 4 %inc = add i32 %6, 1 store i32 %inc, i32* %i, align 4 br label %for.cond for.end:
```

; preds = %for.cond

ret i32 0