[6/7]
entry:
%retval = alloca i32, align 4
%i = alloca i32, align 4
store i32 0, i32* %retval, align 4
store i32 1, i32* %i, align 4
br label %for.cond

```
for.body:
                                                    ; preds = \% for.cond
                              %1 = load i32, i32* %i, align 4
                              %idxprom = zext i32 %1 to i64
 %arrayidx = getelementptr inbounds [100 x i32], [100 x i32]* @_ZL1a, i64 0, i64 %idxprom
                          %2 = load i32, i32* %arrayidx, align 4
                              %3 = load i32, i32* %i, align 4
                             % idxprom 1 = zext i 32 \% 3 to i 64
                                  %add = add i32 %2. 0
                              %4 = load i32, i32* \%i, align 4
                             %idxprom3 = zext i32 %4 to i64
%arrayidx4 = getelementptr inbounds [100 x i32], [100 x i32]* @_ZL1c, i64 0, i64 %idxprom3
                         store i32 %add, i32* %arrayidx4, align 4
                             %5 = load i32, i32* %i, align 4
                                  % sub = sub i32 %5, 1
                            %idxprom5 = zext i32 %sub to i64
%arrayidx6 = getelementptr inbounds [100 x i32], [100 x i32]* @_ZL1c, i64 0, i64 %idxprom5
                         %6 = load i32, i32* %arrayidx6, align 4
                             %7 = load i32, i32* %i, align 4
                             %idxprom7 = zext i32 %7 to i64
%arrayidx8 = getelementptr inbounds [100 x i32], [100 x i32]* @_ZL1a, i64 0, i64 %idxprom7
                          store i32 %6, i32* %arrayidx8, align 4
                                    br label %for.inc
```

```
for.inc: ; preds = %for.body %8 = load i32, i32* %i, align 4 %inc = add i32 %8, 1 store i32 %inc, i32* %i, align 4 br label %for.cond [2/3]
```

; preds = % for .cond