

```
entry:  
  
%retval = alloca 1  
%i.addr = alloca 1  
%j.addr = alloca 1  
%k = alloca 1  
store %i , %i.addr  
store %j , %j.addr  
store 0 , %k  
br %for.cond
```

```
for.cond:  
  
%0 = load %k  
%1 = load %i.addr  
%idxprom = sext %1  
%arrayidx = getelementptr @piecemax , 0 , %idxprom  
%2 = load %arrayidx  
%cmp = icmp %0 , %2  
br %cmp , %for.end , %for.body
```

```
for.body:  
  
%3 = load %i.addr  
%idxprom1 = sext %3  
%arrayidx2 = getelementptr @p , 0 , %idxprom1  
%4 = load %k  
%idxprom3 = sext %4  
%arrayidx4 = getelementptr %arrayidx2 , 0 , %idxprom3  
%5 = load %arrayidx4  
%tobool = icmp %5 , 0  
br %tobool , %if.end , %if.then
```

```
if.then:  
  
%6 = load %j.addr  
%7 = load %k  
%add = add %6 , %7  
%idxprom5 = sext %add  
%arrayidx6 = getelementptr @puzzl , 0 , %idxprom5  
store 1 , %arrayidx6  
br %if.end
```

```
if.end:  
br %for.inc
```

```
for.inc:  
  
%8 = load %k  
%inc = add %8 , 1  
store %inc , %k  
br %for.cond
```

```
for.end:  
  
%9 = load %i.addr  
%idxprom7 = sext %9  
%arrayidx8 = getelementptr @class , 0 , %idxprom7  
%10 = load %arrayidx8  
%idxprom9 = sext %10  
%arrayidx10 = getelementptr @piececount , 0 , %idxprom9  
%11 = load %arrayidx10  
%sub = sub %11 , 1  
%12 = load %i.addr  
%idxprom11 = sext %12  
%arrayidx12 = getelementptr @class , 0 , %idxprom11  
%13 = load %arrayidx12  
%idxprom13 = sext %13  
%arrayidx14 = getelementptr @piececount , 0 , %idxprom13  
store %sub , %arrayidx14  
%14 = load %j.addr  
store %14 , %k  
br %for.cond15
```

```
for.cond15:  
  
%15 = load %k  
%cmp16 = icmp %15 , 511  
br %cmp16 , %for.end25 , %for.body17
```

```
for.body17:  
  
%16 = load %k  
%idxprom18 = sext %16  
%arrayidx19 = getelementptr @puzzl , 0 , %idxprom18  
%17 = load %arrayidx19  
%tobool20 = icmp %17 , 0  
br %tobool20 , %if.then21 , %if.end22
```

```
if.end22:  
br %for.inc23
```

```
for.inc23:  
  
%19 = load %k  
%inc24 = add %19 , 1  
store %inc24 , %k  
br %for.cond15
```

```
if.then21:  
  
%18 = load %k  
store %18 , %retval  
br %return
```

```
return:  
  
%20 = load %retval  
ret %20
```

```
for.end25:  
store 0 , %retval  
br %return
```