

```
entry:
%retval = alloca i32, align 4
%A = alloca [10 x i32], align 16
%B = alloca [10 x i32], align 16
%i = alloca i32, align 4
%j = alloca i32, align 4
%k = alloca i32, align 4
store i32 0, i32* %retval, align 4
%0 = bitcast [10 x i32]* %A to i8*
call void @llvm.memcpy.p0i8.p0i8.i64(i8* align 16 %0, i8* align 16 bitcast
... ([10 x i32]* @_const.main.A to i8*), i64 40, i1 false)
%1 = bitcast [10 x i32]* %B to i8*
call void @llvm.memset.p0i8.i64(i8* align 16 %1, i8 0, i64 40, i1 false)
store i32 37, i32* %k, align 4
store i32 0, i32* %j, align 4
store i32 0, i32* %i, align 4
br label %for.cond
```

```
for.cond:
%2 = load i32, i32* %i, align 4
%cmp = icmp slt i32 %2, 10
br i1 %cmp, label %for.body, label %for.end, !prof !34
```

T	F
---	---

```
for.body:
%3 = load i32, i32* %k, align 4
%mul = mul nsw i32 %3, 2
%4 = load i32, i32* %j, align 4
%idxprom = sext i32 %4 to i64
%arrayidx = getelementptr inbounds [10 x i32], [10 x i32]* %A, i64 0, i64
... %idxprom
%5 = load i32, i32* %arrayidx, align 4
%mul1 = mul nsw i32 %5, 23
%add = add nsw i32 %mul, %mul1
%6 = load i32, i32* %i, align 4
%add2 = add nsw i32 %add, %6
%7 = load i32, i32* %i, align 4
%idxprom3 = sext i32 %7 to i64
%arrayidx4 = getelementptr inbounds [10 x i32], [10 x i32]* %B, i64 0, i64
... %idxprom3
store i32 %add2, i32* %arrayidx4, align 4
%8 = load i32, i32* %i, align 4
%rem = srem i32 %8, 7
%cmp5 = icmp eq i32 %rem, 0
br i1 %cmp5, label %if.then, label %if.end10, !prof !35
```

T	F
---	---

```
if.then:
%9 = load i32, i32* %i, align 4
%rem6 = srem i32 %9, 2
%cmp7 = icmp eq i32 %rem6, 1
br i1 %cmp7, label %if.then8, label %if.else, !prof !36
```

T	F
---	---

```
if.then8:
%10 = load i32, i32* %i, align 4
store i32 %10, i32* %j, align 4
br label %if.end
```

```
if.else:
%11 = load i32, i32* %i, align 4
%add9 = add nsw i32 %11, 1
store i32 %add9, i32* %k, align 4
br label %if.end
```

```
if.end:
br label %if.end10
```

```
if.end10:
%12 = load i32, i32* %i, align 4
%idxprom11 = sext i32 %12 to i64
%arrayidx12 = getelementptr inbounds [10 x i32], [10 x i32]* %B, i64 0, i64
... %idxprom11
%13 = load i32, i32* %arrayidx12, align 4
%call = call i32 (i8*, ...) @printf(i8* noundef getelementptr inbounds ([4 x
... i8], [4 x i8]* @.str, i64 0, i64 0), i32 noundef %13)
br label %for.inc
```

```
for.inc:
%14 = load i32, i32* %i, align 4
%inc = add nsw i32 %14, 1
store i32 %inc, i32* %i, align 4
br label %for.cond, !llvm.loop !37
```

```
for.end:
ret i32 0
```

CFG for 'main' function