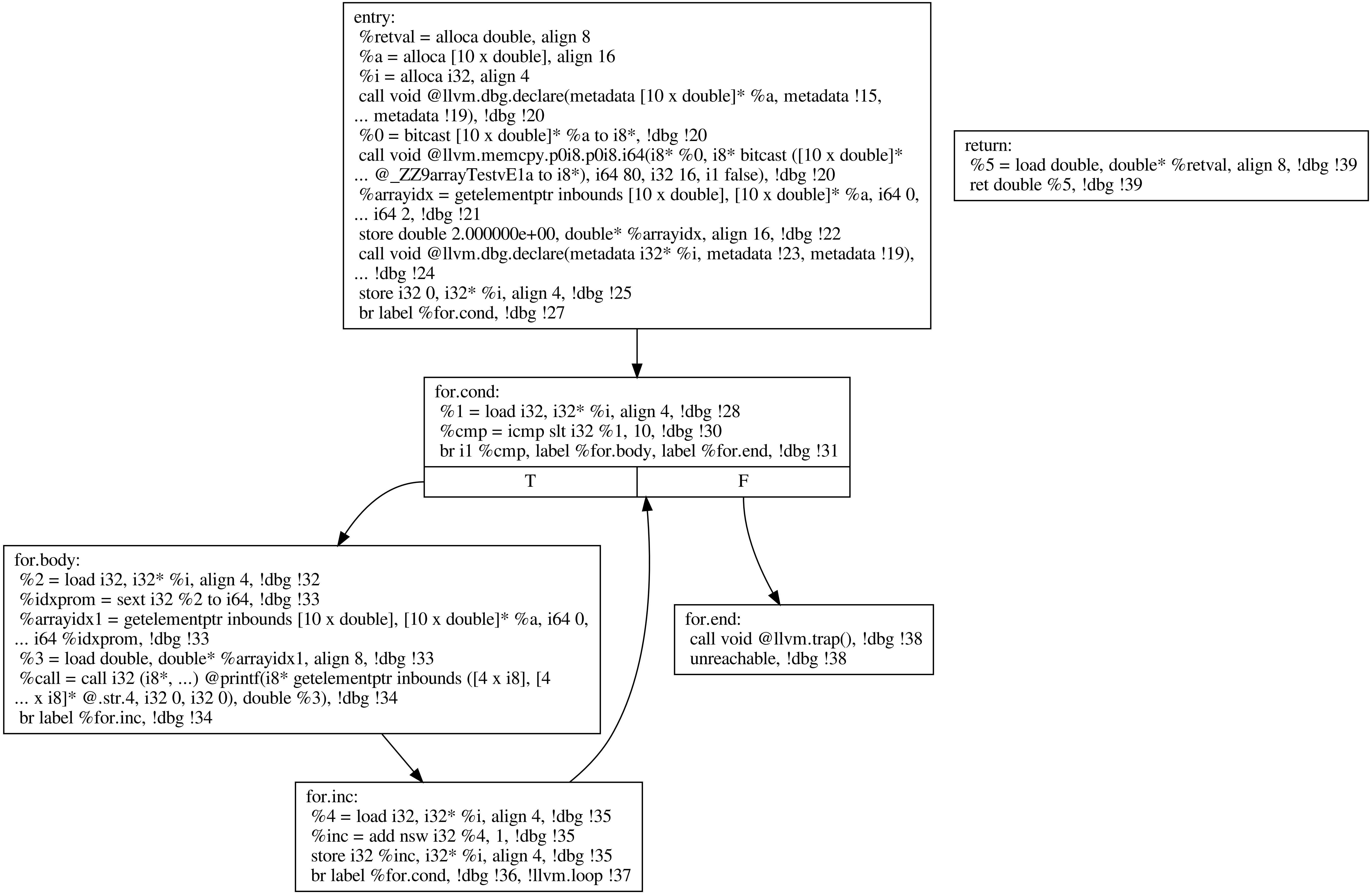

```
entry:
 %retval = alloca double, align 8
 %a = alloca [10 x double], align 16
 %i = alloca i32, align 4
 call void @llvm.dbg.declare(metadata [10 x double]* %a, metadata !15,
... metadata !19), !dbg !20
 %0 = bitcast [10 x double]* %a to i8*, !dbg !20
 call void @llvm.memcpy.p0i8.p0i8.i64(i8* %0, i8* bitcast ([10 x double]*
... @_ZZ9arrayTestvE1a to i8*), i64 80, i32 16, i1 false), !dbg !20
 %arrayidx = getelementptr inbounds [10 x double], [10 x double]* %a, i64 0,
... i64 2, !dbg !21
 store double 2.000000e+00, double* %arrayidx, align 16, !dbg !22
 call void @llvm.dbg.declare(metadata i32* %i, metadata !23, metadata !19),
... !dbg !24
 store i32 0, i32* %i, align 4, !dbg !25
 br label %for.cond, !dbg !27
```

```
return:
 %5 = load double, double* %retval, align 8, !dbg !39
 ret double %5, !dbg !39
```

```
for.cond:
 %1 = load i32, i32* %i, align 4, !dbg !28
 %cmp = icmp slt i32 %1, 10, !dbg !30
 br i1 %cmp, label %for.body, label %for.end, !dbg !31
```

```
for.body:
 %2 = load i32, i32* %i, align 4, !dbg !32
 %idxprom = sext i32 %2 to i64, !dbg !33
 %arrayidx1 = getelementptr inbounds [10 x double], [10 x double]* %a, i64 0,
... i64 %idxprom, !dbg !33
 %3 = load double, double* %arrayidx1, align 8, !dbg !33
 %call = call i32 (i8*, ...) @printf(i8* getelementptr inbounds ([4 x i8], [4
... x i8]* @.str.4, i32 0, i32 0), double %3), !dbg !34
 br label %for.inc, !dbg !34
```

```
for.end:
 call void @llvm.trap(), !dbg !38
 unreachable, !dbg !38
```

```
for.inc:
 %4 = load i32, i32* %i, align 4, !dbg !35
 %inc = add nsw i32 %4, 1, !dbg !35
 store i32 %inc, i32* %i, align 4, !dbg !35
 br label %for.cond, !dbg !36, !llvm.loop !37
```

CFG for '_Z9arrayTestv' function