```
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
store i32 0, i32* %retval, align 4
\%0 = \text{bitcast} [10 \times i32] * \% A \text{ 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 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
                                                     F
                         90.91%
                                                                                                                      9.09%
                                        for.body:
                                         %3 = load i32, i32* %j, align 4
                                         %idxprom = sext i32 %3 to i64
                                         %arrayidx = getelementptr inbounds [10 x i32], [10 x i32]* %A, i64 0, i64
                                        ... %idxprom
                                        %4 = load i32, i32* %arrayidx, align 4
                                         %mul = mul nsw i32 %4, 13
                                         %add = add nsw i32 %mul, 4
                                        %5 = load i32, i32* %i, align 4
                                        %add1 = add nsw i32 %add, %5
                                                                                                                             for.end:
                                         \%6 = load i32, i32*\%i, align 4
                                                                                                                              ret i32 0
                                         %idxprom2 = sext i32 \%6 to i64
                                         % \operatorname{arrayidx} 3 = \operatorname{getelementptr} in \text{bounds} [10 \times i32], [10 \times i32] * \% B, i64 0, i64
                                        ... %idxprom2
                                        store i32 %add1, i32* %arrayidx3, align 4
                                        %7 = load i32, i32* %i, align 4
                                         %rem = srem i32 \%7, 8
                                         %cmp4 = icmp eq i32 %rem, 0
                                        br i1 %cmp4, label %if.then, label %if.end, !prof!35
                                                                                                    F
                                                             20.00%
                                                  if.then:
                                                  \%8 = \text{load i}32, i32*\%i, align 4
                                                                                                    80.00%
                                                  store i32 %8, i32* %j, align 4
                                                  br label %if.end
                                        if.end:
                                         %9 = load i32, i32* %i, align 4
                                         %idxprom5 = sext i32 \%9 to i64
                                         %arrayidx6 = getelementptr inbounds [10 x i32], [10 x i32]* %B, i64 0, i64
                                        ... %idxprom5
                                        %10 = load i32, i32* %arrayidx6, align 4
                                         %call = call i32 (i8*, ...) @printf(i8* noundef getelementptr inbounds ([4 x
                                        ... i8], [4 x i8]* @.str, i64 0, i64 0), i32 noundef %10)
                                         br label %for.inc
                   for.inc:
                   %11 = load i32, i32* %i, align 4
```

%inc = add nsw i32 %11, 1 store i32 %inc, i32\* %i, align 4 br label %for.cond, !llvm.loop !36

CFG for 'main' function