```
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 = \text{bitcast } [10 \text{ x } i32] * \%B \text{ 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
                                                                %2 = load i32, i32* %j, align 4
                                                                %idxprom = sext i32 %2 to i64
                                                                %arrayidx = getelementptr inbounds [10 x i32], [10 x i32]* %A, i64 0, i64
                                                                ... %idxprom
                                                                %3 = load i32, i32* %arrayidx, align 4
                                                                %mul = mul nsw i32 %3, 13
                                                                %add = add nsw i32 %mul, 4
                                                                %var = alloca i32, align 16
                                                                store i32 %add, i32* %var, align 4
                                                                br label %for.cond
                                                                          for.cond:
                                                                           %4 = load i32, i32* %i, align 4
                                                                           %cmp = icmp slt i32 %4, 10
                                                                           br i1 %cmp, label %for.body, label %for.end, !prof !34
                   for.body:
                   %5 = Ĭoad i32, i32* %i, align 4
                    %fix = load i32, i32* %var, align 4
                    %add1 = add nsw i32 \%fix, \%5
                    %6 = load i32, i32* %i, align 4
                    %idxprom2 = sext i32 %6 to i64
                    % \operatorname{arrayidx} 3 = \operatorname{getelementptr} in \text{bounds} [10 \times i32], [10 \times i32] * \% B, i64 0, i64
                                                                                                                 for.end:
                    ... %idxprom2
                                                                                                                 ret i32 0
                   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
%8 = load i32, i32* %i, align 4
\%9 = \text{sext i} 32 \%8 \text{ to i} 64
%10 = getelementptr inbounds [10 x i32], [10 x i32]* %A, i64 0, i64 %9
%11 = load i32, i32* %10, align 4
%12 = mul nsw i32 %11, 13
%13 = add nsw i32 %12, 4
store i32 %13, i32* %var, align 4
br label %if.end
                   if.end:
                    %14 = load i32, i32* %i, align 4
                    %idxprom5 = sext i32 %14 to i64
                    %arrayidx6 = getelementptr inbounds [10 x i32], [10 x i32]* %B, i64 0, i64
                   ... %idxprom5
                    %15 = 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 %15)
                    br label %for.inc
                                                                for.inc:
                                                                 %16 = load i32, i32* %i, align 4
                                                                 %inc = add nsw i32 %16, 1
                                                                 store i32 %inc, i32* %i, align 4
                                                                 br label %for.cond, !llvm.loop !36
```

if.then:

CFG for 'main' function