```
entry:
                                              %retval = alloca i32, align 4
                                              %A = alloca [10 \times 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] \% \text{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 \times 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
                                                                      Т
                                                                                                     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, 23
%5 = load i32, i32* %i, align 4
%add = add nsw i32 %mul, %5
                                                                                                 for.end:
%6 = load i32, i32* %i, align 4
                                                                                                 ret i32 0
%idxprom1 = sext i32 \%6 to i64
% \operatorname{arrayidx2} = \operatorname{getelementptr} \operatorname{inbounds} [10 \times i32], [10 \times i32] * \%B, i64 0, i64
... %idxprom1
store i32 %add, i32* %arrayidx2, align 4
\%7 = \text{load i32}, \text{i32* \%i, align 4}
%rem = srem i32 \%7, 8
%cmp3 = icmp eq i32 %rem, 0
br i1 %cmp3, label %if.then, label %if.else, !prof !35
                                                              F
                    |20.00%
                                                               80.00%
                                            if.else:
    if.then:
                                             \%9 = \text{load i} 32, i 32*\%i, align 4
     \%8 = \text{load i} 32, i 32* \%i, align 4
                                             %add4 = add nsw i32 \%9.1
     store i32 %8, i32* %j, align 4
                                             store i32 %add4, i32* %j, align 4
     br label %if.end
                                             br label %if.end
if.end:
%10 = load i32, i32* %i, align 4
%idxprom5 = sext i32 %10 to i64
% \operatorname{arrayidx} 6 = \operatorname{getelementptr} in bounds [10 x i32], [10 x i32] * %B, i64 0, i64
... %idxprom5
%11 = 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 %11)
br label %for.inc
                                              for.inc:
                                               %12 = load i32, i32* %i, align 4
                                               %inc = add nsw i32 %12, 1
                                               store i32 %inc, i32* %i, align 4
                                               br label %for.cond, !llvm.loop !36
```

CFG for 'main' function