entry: %retval = alloca i32, align 4 %argc.addr = alloca i32, align 4 %argv.addr = alloca i8**, align 8 %a = alloca i32, align 4 %sum = alloca i32, align 4 %i = alloca i32, align 4 %c = alloca i32, align 4store i32 0, i32* %retval, align 4 store i32 %argc, i32* %argc.addr, align 4 store i8** %argv, i8*** %argv.addr, align 8 store i32 0, i32* %a, align 4 store i32 0, i32* %sum, align 4 store i32 0, i32* %i, align 4 br label %for.cond for.cond: %0 = load i 32, i 32*% i, align 4%1 = load i32, i32* %argc.addr, align 4 %cmp = icmp slt i32 %0, %1 br il %cmp, label %for.body, label %for.end F Т for.end: %5 = load i32, i32* %a, align 4for.body: %6 = load i32, i32* %sum, align 4 %2 = load i 32, i 32*% i, align 4%call = call i32 @foo(i32 %5. i32 %6) %3 = load i32, i32* %sum, align 4store i32 %call, i32* %c, align 4 %add = add nsw i32 %3, %2 %7 = load i 32, i 32* %c, align 4store i32 %add, i32* %sum, align 4 %call1 = call i32 (i8*, ...) @printf(i8* getelementptr inbounds ([5 x i8], br label %for.inc ... $[5 \times i8]$ * @.str, i320, i320), i32%7) ret i32 0 for.inc: %4 = load i32, i32* %i, align 4%inc = add nsw i32 %4, 1 store i32 %inc, i32* %i, align 4 br label %for.cond

CFG for 'main' function