

# 1. 程序实现功能

补充完整对函数定义与调用的翻译：允许数组作为函数的参数，其中数组仅包含一维数组。

## 2. 如何实现

1. 在BaseSymbol类中添加两个boolean类型的属性：isFParam，用于记录是否是形参；isArray，用于记录是否是数组。
2. visit函数定义：定义函数形参类型时，如果是数组，则其类型为LLVMPointerType(i32Type, 0)；并使用以下方法为数组形参分配空间、添加到当前作用域：

```
LLVMValueRef pointer = LLVMBuildAlloca(builder, LLVMPointerType(i32Type, 0),  
/*pointerName:String*/s);  
LLVMBuildStore(builder, n, pointer);  
baseSymbol = new BaseSymbol(s, pointer);  
baseSymbol.setArray(true);  
baseSymbol.setFParam(true);
```

3. visitLVal：使用以下方法取出形参数组中的某个值：

```
String name = ctx.IDENT().getText();  
Symbol symbol = currentScope.resolve(name);  
LLVMValueRef pointer = LLVMBuildLoad(builder, vectorPointer, name);  
PointerPointer valuePointer = new PointerPointer(new LLVMValueRef[]{value});  
LLVMValueRef res = LLVMBuildGEP(builder, pointer, valuePointer, 1, "res");  
return LLVMBuildLoad(builder, res, name);
```

如果不是取出某个值，而是将整个形参数组作为实参再次传递给函数调用的话，这时候其实传递的是指向数组第一个值的指针，所以直接return LLVMBuildLoad(builder, pointer, "res")。

## 3. 印象深刻的bug

主要错的有normaltest0、1、4、9，而这4个用例错误的原因都是因为，每次形参数组被重新赋值以后，我都会创建一个新的symbol并添加到符号表，以覆盖原来的symbol，而创建时我忘记把它的isFParam和isArray设置为true了，导致并没有进行形参数组相关的操作。修改以后就正确了。