CSci 340, Spring 2003, Project 4

This project is due *Friday, April 4* at 11:20am. Because this assignment is relatively short, it is worth only 35 points. I will award everybody the remaining 25 points free.

In this class, we'll define an expression type in Haskell, similar to the Context type of Assignment 3.

```
type Context = String -> Integer
type Expr = Context -> Integer
```

Note that the Expr type is a function from a Context to an Integer; if I give an Expr value a Context as a parameter, it should return the Integer value of the expression for that context.

Your job is to construct the following functions for generating Expr values.

```
identExpr :: String -> Expr
intExpr :: Integer -> Expr
addExpr :: Expr -> Expr -> Expr
subExpr :: Expr -> Expr -> Expr
multExpr :: Expr -> Expr -> Expr
```

Note the similarity to Project 2.

Using these functions, I should be able to define expression values.

```
e = addExpr (intExpr 4) (identExpr "j")
c = addIdent (addIdent emptyContext "i" 8) "j" 31
```

The expression `e` would represent the expression $4 + j$. With these defined, the expression `e c` should return 35, since `c` associates the value 31 with j.

The `emptyContext` and `addIdent` functions are from Assignment 3. Here are the correct definitions.

```
emptyContext = \s -> error ("unknown identifier " ++ s)
addIdent c id val = \s -> if s == id then val else c s
```

Your submitted solution need only contain your individual functions, on paper, defined for this assignment.