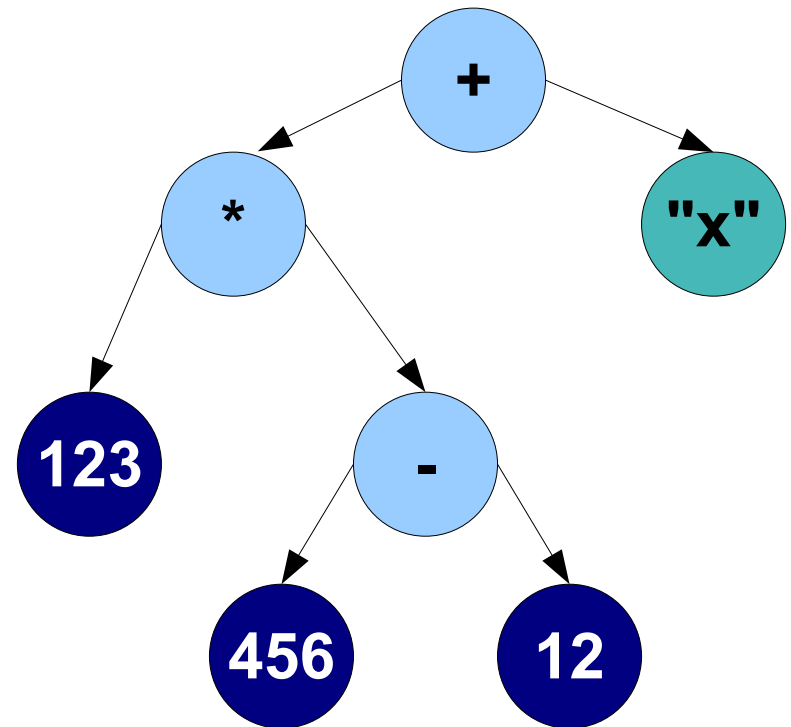


# Exercice

- Représentation d'une expression arithmétique

```
typedef enum {  
    TYPE_CONST,  
    TYPE_VAR,  
    TYPE_ADD,  
    TYPE_SUBTRACT,  
    TYPE_MULTIPLY,  
    TYPE_DIVIDE,  
} tree_node_type_t;  
  
typedef struct tree_node_t {  
    struct tree_node_t * left;  
    struct tree_node_t * right;  
    tree_node_type_t      type;  
    union {  
        double   value;  
        char     * variable;  
    };  
} tree_node_t;
```



# Exercice

- Représentation d'une expression arithmétique
  - Usage typique

```
int main() {  
    /*  
     * The following expression is (((123)*(456-12))+(x))  
     */  
    tree_node_t * root=  
        ADD(MUL(CONST(123), SUB(CONST(456), CONST(12))), VAR("x"));  
  
    printf("Expr  : ");  
    tree_node_print(root);  
    printf("\n");  
    printf("Result: %f\n", tree_node_eval(root));  
    tree_node_free(&root);  
  
    return 0;  
}
```

# Exercice

- Représentation d'une expression arithmétique
  - Question subsidiaire: comment modifier la structure pour permettre de travailler sur des nombres complexes ? sur des matrices ?