

```
/* ideas for useful functions for Project 2 */
```

```
char getCorrespondingSymbol ( char ch)
{
    if ( ch ==n ')' ) return '(';
    if ( ch ==n '(' ) return ')';
    if ( ch ==n '<' ) return '>';
    ....
}
```

```
// variation on above is array of characters
char getCorrespondingSymbolV2 ( char ch)
{
    char symArray[8] = { '(', '<', '{', '[', ')', '>', '}', ']' }

    int i;
    for ( i = 0 ; i < 8 ; i++ )
        if ( symArray[i] == ch )
            return symArray [ (i + 4) % 8 ] ;

    /* return null character if parameter does not have corresponding symbol
    return '\0';
}
```

```
int isOpenSymbol ( char ch )
{
    if ( ch == '(' || ch = '<' || ... )
        return TRUE;
    else
        return FALSE;
}
```

```
int isCloseSymbol ( char ch)
{
    if ( ch == ')' || ch = '>' || ... )
        return TRUE;
    else
        return FALSE;
}
```

Typical if statement to check if true:

```
#define TRUE 1
```

```
if ( boolValue == TRUE )
```

Issue: any non-zero value is considered true. Sometimes library functions return a non-one value for true.

A better statement is

```
if ( boolValue != FALSE )
```

```
/* typical C boolean set-up */
```

```
#define TRUE 1
```

```
#define FALSE 0
```

From <https://www.cs.uic.edu/pub/CS211/LabsF17/lab4b.c>

```
if ( .... )  
    return TRUE;  
else  
    return FALSE;
```

BEWARE OF INVALID #define statements

Douglas Adams:

The answer to Life, the Universe and Everything is 42.

So what is the question?

What do you get when you multiple 9 by 6?

Wait!! The is wrong. (Or just the twisted humor of a clever author)

One explanation - using faulty C #define statements:

```
#define NINE 1 + 8
```

```
#define SIX 5 + 1
```

```
int answer;
```

```
answer = NINE * SIX ;
```

```
printf ("The answer to Life, the Universe and Everything is %d\n", answer);
```

The statement

```
answer = NINE * SIX ;
```

becomes the following after the preprocessor runs

```
answer = 1 + 8 * 5 + 1 ;
```

Since multiplication has high precedence than addition, the fully parenthesized version is

```
answer = 1 + (8 * 5) + 1;
```

Which evaluates to

```
answer = 42
```