

```

/* chad c d clark < clarkch @ cpsc . ucalgary . ca >
*
* cpsc 411      lec ??
* winter 2002  lab 02
*
* assignment #1 - a first stab.
*
* file: asltree.h
* purpose: defines the tree structures to be used in the syntax tree.
*
*
*/

#define IF_NODE          501
#define WHILE_NODE      502
#define DO_NODE         503
#define ASSIGN_NODE     504
#define READ_NODE       505
#define PRINT_NODE      506
#define BEGIN_NODE     507
#define STMTLIST_NODE  508
#define ADD_NODE        509
#define SUB_NODE        510
#define MUL_NODE        511
#define DIV_NODE        512
#define ID_NODE         513
#define NUM_NODE        514

struct stree_node {

    /* type of node this is */
    int type;

    /* the next argument for a parent node */
    struct stree_node * sibling;
    /* the sub-tree link */
    struct stree_node * child;

    /* numeric value of this node */
    int numval;
    /* string value of this node (used for identifier names) */
    char * idval;
};

/* functions for stmt -> rules *****/

struct stree_node * makeIFnode(struct stree_node *if_expr,
                              struct stree_node *true_stmt,
                              struct stree_node *false_stmt);

struct stree_node * makeWHILEnode(struct stree_node *while_expr,
                                  struct stree_node *do_stmt);

struct stree_node * makeDONode(struct stree_node *do_stmt,
                               struct stree_node *while_expr);

struct stree_node * makeASSIGNnode(struct stree_node *id_node,

```

```
        struct stree_node *an_expr);

struct stree_node *makeREADnode(struct stree_node *id_node);

struct stree_node *makePRINTnode(struct stree_node *an_expr);

struct stree_node *makeBEGINnode(struct stree_node *slist);

/* functions for stmtlist -> rules *****/
struct stree_node *makeSTMTLISTnode(struct stree_node *a_stmt,
        struct stree_node *a_stmtlist);

/* functions for expr-> rules *****/
struct stree_node *makeADDnode(struct stree_node *a_term,
        struct stree_node * right_term);

struct stree_node *makeSUBnode(struct stree_node *a_term,
        struct stree_node * right_term);

/* functions for term -> rules *****/
struct stree_node *makeMULnode(struct stree_node *a_factor,
        struct stree_node * right_factor);

struct stree_node *makeDIVnode(struct stree_node *a_factor,
        struct stree_node * right_factor);

/* functions for factor -> rules *****/
struct stree_node *makeIDnode(char *ident);

struct stree_node *makeNUMnode(char *num);

/* function to print out the syntax tree *****/
void print_stree(struct stree_node *node, int spaces);

/* function to find a type of node in the tree *****/
int find_node(int type, struct stree_node *node);

/* function to recursively delete a tree *****/
void delete_stree(struct stree_node *node);
```