Quiz 10 Solution
yacc Example
Project Comments

Quiz 10 Solution

```
%{
    #include "semantics.h"
    #include "y.tab.h"
%}

digit   [0-9]

%%
{digit}+            {return IntLit;}
\,                  {return ',';}
\;                  {return ';';}
\t                  {}
\n                  {}
[ ]                  {}


%%

int yywrap() {
    return 1;
}
```

Quiz 10 Solution

```
%{

extern int yylex();
extern int yyerror(char *);
extern char * yytext;
#include <stdio.h>
#include <stdlib.h>
#include "semantics.h"
%}

%union {
    int val;
    Node * Node;
}

%type <val> Element
%type <Node> List

%token IntLit
```

Quiz 10 Solution

```
%%

Prog    : Lists                 { };
Lists   : SumList Lists         { };
Lists   :                       { };
SumList : List ';'              {finish($1);};
List    : Element               {$$ = add($1, NULL);};
List    : Element ',' List      {$$ = add($1, $3);};
List    :                       { };
Element : IntLit                {$$ = atoi(yytext);};

%%

int yyerror(char *s) {
    printf("%s Failure\n", yytext);
    exit(1);
}

int main(int argc, char * argv[]) {
    yyparse();
}
```

Quiz 10 Solution

```c
typedef struct Node {
    int num;
    struct Node * next;
} Node;

Node * add(int x, Node *h);
void finish(Node * h);
```

Quiz 10 Solution

```c
#include "semantics.h"
#include <stdlib.h>
#include <stdio.h>

Node * add(int x, Node *h) {
  Node * h1 = (Node *) malloc(sizeof(Node));
  h1->num = x;
  h1->next = h;
  return h1;
}
void finish(Node * h) {
  Node * temp;
  int sum;
  if (h == NULL) {
     printf("The list is empty\n");
     return;
  }
  printf("The sum of %d",h->num);
  sum = h->num;
  temp = h->next;
  while (temp != NULL) {
    printf(", %d", temp->num);
    sum = sum + temp->num;
    temp = temp->next;
  }
  printf(" is %d\n", sum);
}
```

Actions in Productions

Input: `((()))()(());`

Output: `(1(2(3)))(1)(1(2))`

Actions in Productions

```
%{
#include "y.tab.h"
%}

Letter [A-Za-z]
Digit [0-9]

%%
\(   { return '(';}
\)   { return ')';}
\;   { return ';';}
.  {}

%%

int yywrap() {
    return 1;
}
```

Actions in Productions

```
%{
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

extern int yylex(); /* The next token function. */
extern char *yytext;   /* The matched token text.  */
extern int yyparse();
char *makeParenStr(int, char*, char*);
int yyerror(char *s);

int depth;

%}
%union {
  intnum;
  char * string;
}

%type <string> Parens;
```

Actions in Productions

```
%%

Prog   : Parens                                                    {printf(“%s\n", $1);}
Parens  : '('{$<num>$ =++ depth;} Parens')'{--depth;}Parens  { $$ = makeParenStr($<num>2, $3, $6);} ;
Parens  :                                                          { $$ = "";} ;

%%
```

Actions in Productions

```
int main(int argc, char *argv[]){
  yyparse();
}

char * makeParenStr(int x, char *p1, char *p2) {
    char *temp = (char *) malloc(10+strlen(p1)+strlen(p2));
    sprintf(temp, "(%d%s)%s", x,p1,p2);
    return temp;
}

int yyerror(char *s) {
  printf("err: \"%s\" yytext: \"%s\"\n",s,yytext);
  exit(1);
}
```

Project

- You must do the items in the order shown on the previous slides. For example before you do arrays you must have integer expressions, integer I/O and the control structures implemented. New features must work with the old features. For example when you implement arrays you must be able to read and print array elements.

- I will give you test programs written in C. You can translate these programs to your language (usually only the I/O will need to be changed)

-

Project Submission

- To submit your project you will send me one zip file. The file will contain your source code, a README file listing the parts of the project you completed, the test programs you ran and files containing the results (copy of the terminal window output of the test programs). There must be one file for each test program result and the name of the file must be the test program name .txt. For example the result of test program t1 must be t1.txt

Project Due Dates

- 11:59 PM Friday December 11 (5 bonus points)
- 11:59 PM Friday December 18