

## تقرير مشروع مادة تصميم مترجمات

### ● أبجدية اللغة ( مفردات اللغة ):

- 1- الكلمات المفتاحية : if – else شرط أن تكتب بحالة الأحرف الصغيرة lower case.
- 2- التعريف عن متحولات وفق قواعد التسمية ( مجموعة من الأحرف الإنكليزية والأرقام [0 – 9] وإشارة \_  
شرط ألا يبدأ اسم المتحول برقم ).
- 3- الأرقام الصحيحة والأرقام الحقيقية ( وتقبل ذات الفاصلة العائمة ).
- 4- \$\$ تشير إلى نهاية التنفيذ.
- 5- التعليقات السطرية ومتعددة الأسطر.
- 6- الرموز : { } وتشير إلى بداية ونهاية مجموعة التعليمات.  
( ) وتشير إلى بداية ونهاية الشرط.  
; وتشير إلى نهاية التعليمة.  
^ + - \* / تشير إلى العمليات الحسابية الأربع بالإضافة لعملية الرفع إلى قوة.  
= تشير إلى عملية الإسناد.  
& | تشير إلى العمليات المنطقية OR AND.  
عملیات المقارنة : == المساواة.  
!= عدم المساواة.  
< , > , <= , >= مقارنة الأعداد والتعابير الحسابية.

### كل ما يكتب عدا ما سبق يعتبر خطأ مفرداتي.

- يتم تجاهل الفراغات والأسطر الجديدة.
- في حال وجود أي خطأ مفرداتي فإن المترجم يتوقف ولا يكمل.

### ● التعليمات المقبولة ( قواعدياً ):

يوجد ثلاث أنواع من التعليمات :

1- تعليمة الشرط if : ويكون لها الشكل التالي :

```
if ( cond )
{
    stmts
}
```

2- تعليمة الشرط if - else : ويكون لها الشكل التالي :

```
if ( cond)
```

```

{
stmts
}
else
{
stmts
}

```

3- تعليمة الإسناد = ويكون لها الشكل التالي :

متحول : Identifier      Identifier = exp ;

تعبير حسابي : exp

- stmts : تعبر عن مجموعة تعليمات متتالية وكل تعليمة ممكن أن تكون أحد التعليمات الثلاث السابقة.
- لا قيد في مكان التعليمات طالما تحقق شكل التعليمة ( يمكن بدء البرنامج بتعليمة إسناد ).
- عند تقييم الشرط لا يتم الاستفادة من النتيجة في تغيير سياق التنفيذ في البرنامج وإنما يتم المرور على كافة التعليمات للتأكد من سلامتها من الأخطاء القواعدية.
- أولويات العمليات هي على الترتيب :
  1. الأقواس ( ) تعطي الأولوية لما في داخلهما.
  2. إشارة القوة ^ وتجميعها يميني.
  3. إشارتي الضرب \* والقسمة / وتجميعهما يساري.
  4. إشارتي الجمع + والطرح - وتجميعهما يساري.
  5. إشارات المقارنة == , != , >= , <= , > , < وهي لا تقبل التجميع.
  6. العملية المنطقية & وتجميعها يساري.
  7. العملية المنطقية | وتجميعها يساري.
- يقبل قواعديا القيام بعمليات الإسناد المتتالي.
- يقبل قواعديا القيام بعمليات حسابية مع المتحولات.

وسنعرض لكم ما كُتب في الملفين 1. & y.

1. compil.1 :

```

% {
#include <stdio.h>
#include <stdlib.h>
#include "y.tab.h"
extern int yylval;
% }

```

%%

"if"	{ return IF;}
"else"	{ return ELSE;}
"\$\$"	{ return END;}
[ -+ ]? ([0-9]* \. ? [0-9] +   [0-9] + \. [0-9]* ) ([eE] [+-]? [0-9] + ) ?	{ yy1val = atoi(yytext); return NUMBER; }
[A-Za-z_][A-Za-z_0-9]*	{ return Identifier;}
"+"	{ return ADD; }
"_"	{ return SUB; }
"*"	{ return MUL;}
"/"	{ return DIV; }
"^"	{ return POWER;}
"&"	{ return AND;}
" "	{ return OR; }
"="	{ return ('=');}
">"	{ return ('>');}
"<"	{ return ('<');}
"<="	{ return LE_OP; }
">="	{ return GE_OP; }
"=="	{ return EQ_OP; }
"!="	{ return NE_OP; }
"{"	{ return ('{'); }
"}"	{ return ('}'); }
"("	{ return ('(');}
")"	{ return (')');}
";"	{ return (';');}
[ \t ]*	{ }
\n	{ }
"//".*	{ /*ignore Comment */ }
"/*"([ ^* ]*\^[ / ])*"*/"	{ /*ignore Multi Line Comment */ }
.	{ printf("*****mystery*****"); }

%%

int yywrap(){ return 1;}

: compile.y .2

```

% {
    #include <stdio.h>
    #include <stdlib.h>
% }

/* declare tokens */

%token IF ELSE
%token NUMBER Identifier
%token ADD SUB MUL DIV OR AND POWER
%token LE_OP GE_OP EQ_OP NE_OP END

%%

program : stmts END {printf("Accepted");}

stmts:   stmts stmt
        |stmts assign
        |
;

stmt:
    IF '(' cond ')' '{' stmts '}' ELSE '{' stmts '}'
    |IF '(' cond ')' '{' stmts '}'

;

assign:
    Identifier '=' assign
    |Identifier '=' exp ';'

```

```

;
cond: cterm|
      cond OR cterm
;
cterm: cfact|
      cterm AND cfact
;
cfact:
      exp compare exp
      |exp
;
compare:
      '<'
      |'>'|LE_OP|GE_OP|EQ_OP|NE_OP
;
exp: term
   | exp ADD term { }
   | exp SUB term { }
;
term: factor
     | term MUL factor { }
     | term DIV factor { }
;
factor: fact POWER factor
       |fact
;
fact: NUMBER

```

```

|Identifier
|('exp') { }

;

%%

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

yyerror(char *s)
{
    fprintf(stderr, "error: %s\n", s);
}

```

مثال :

```

if(5 < 7 )
{
    d = 8 * 9 ;
    if(5 & 8)
    {
        b=j = 8 ;
    }
}

```

```
    else
    {
        s= 7;
    }
    if(u)
    {
        a = r = 5;
    }
    if(h <= y)
    {}
}
```