(1)

TIKRIT UNIVERSITY

COLLEGE OF COMPUTER SCIENCE AND MATHEMATICS

DEPARTMENT OF COMPUTER SCIENCE

SUBJECT OF COMPILER1

DATE OF ISSUE: 2024 - 2025

CLASS: 3TH STAGE

SEMESTER 1

LAB-NO. : 4

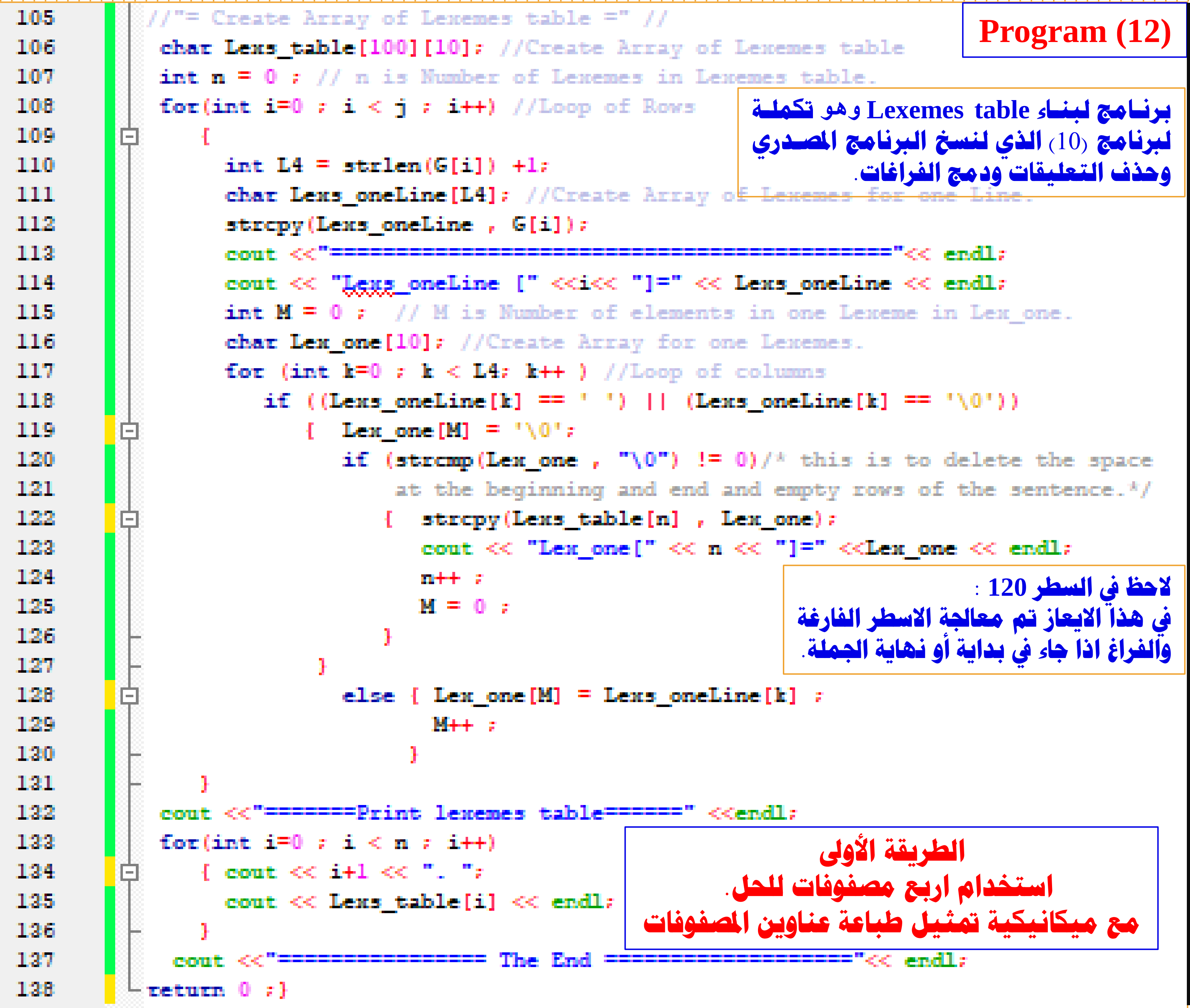
PREPARED BY

Lecturer:
Mohanad Dawood Al-Roomi

&

Assistant Lecturer:
Luay Ibrahim Klalif

Program (12)

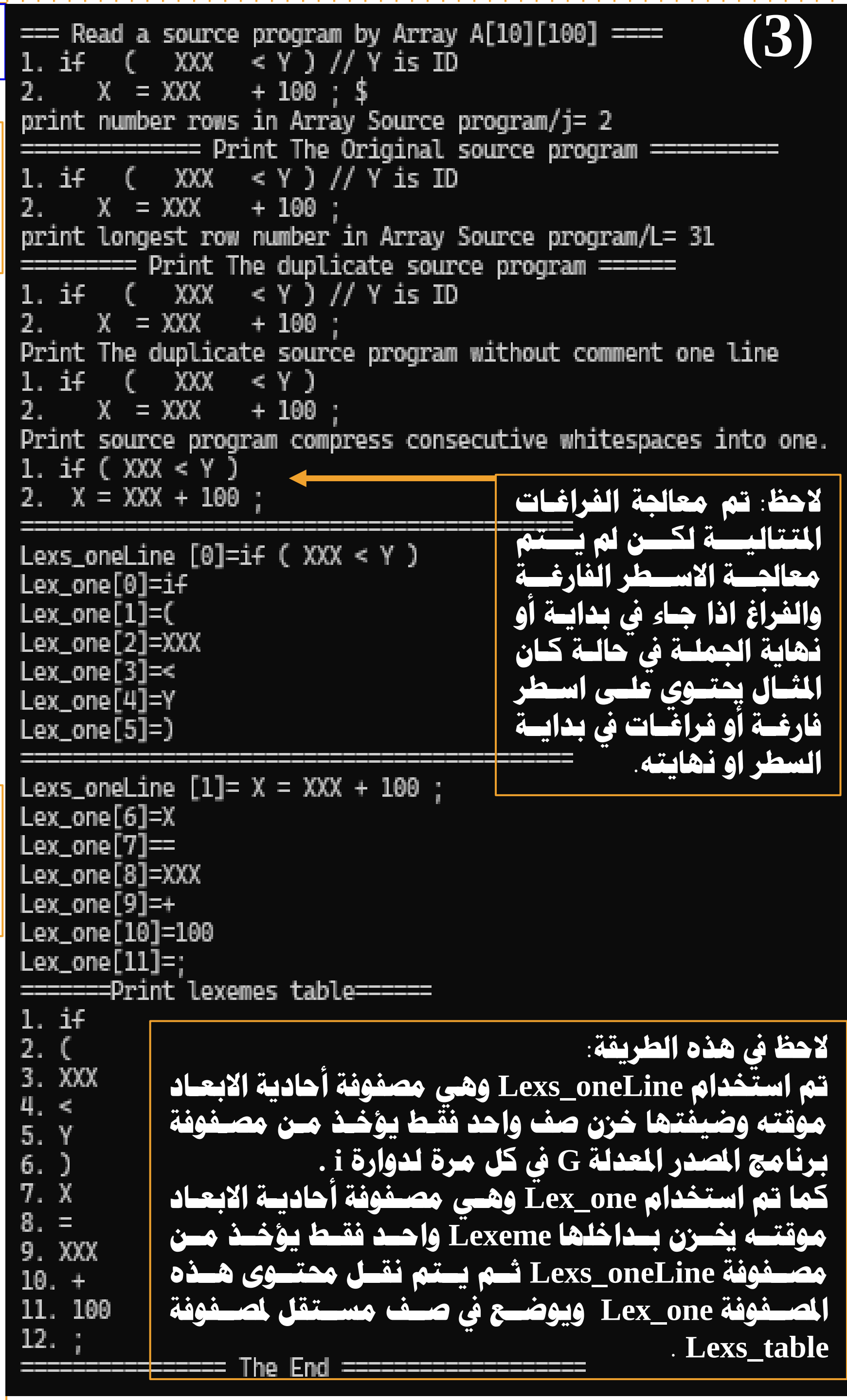


```
105  //"= Create Array of Lexemes table =" //
106   char Lexs_table[100][10]; //Create Array of Lexemes table
107   int n = 0 ; // n is Number of Lexemes in Lexemes table.
108   for(int i=0 ; i < j ; i++) //Loop of Rows
109       {
110          int L4 = strlen(G[i]) +1;
111          char Lexs_oneLine[L4]; //Create Array of Lexemes for one Line.
112          strcpy(Lexs_oneLine , G[i]);
113          cout <<"=========================================="<< endl;
114          cout << "Lexs_oneLine [" <<i<< "]=" << Lexs_oneLine << endl;
115          int M = 0 ;  // M is Number of elements in one Lexeme in Lex_one.
116          char Lex_one[10]; //Create Array for one Lexemes.
117          for (int k=0 ; k < L4; k++ ) //Loop of columns
118             if ((Lexs_oneLine[k] == ' ') || (Lexs_oneLine[k] == '\0'))
119                {  Lex_one[M] = '\0';
120                   if (strcmp(Lex_one , "\0") != 0)/* this is to delete the space
121                      at the beginning and end and empty rows of the sentence.*/
122                      {  strcpy(Lexs_table[n] , Lex_one);
123                         cout << "Lex_one[" << n << "]=" <<Lex_one << endl;
124                         n++ ;
125                         M = 0 ;
126                      }
127                }
128                   else { Lex_one[M] = Lexs_oneLine[k] ;
129                          M++ ;
130                        }
131       }
132   cout <<"=======Print lexemes table======" <<endl;
133   for(int i=0 ; i < n ; i++)
134      { cout << i+1 << ". ";
135        cout << Lexs_table[i] << endl;
136      }
137    cout <<"================ The End ===================="<< endl;
138  return 0 ;}
```

برنامج لبناء Lexemes table وهو تكملة لبرنامج (10) الذي لنسخ البرنامج المصدري وحذف التعليقات ودمج الفراغات.

لاحظ في السطر 120 :
في هذا الايعاز تم معالجة الاسطر الفارغة والفراغ اذا جاء في بداية أو نهاية الجملة.

الطريقة الأولى
استخدام اربع مصفوفات للحل.
مع ميكانيكية تمثيل طباعة عناوين المصفوفات



```
=== Read a source program by Array A[10][100] ====
1. if    (    XXX    < Y ) // Y is ID
2.     X  = XXX     + 100 ; $
print number rows in Array Source program/j= 2
============== Print The Original source program =========
1. if    (    XXX    < Y ) // Y is ID
2.     X  = XXX     + 100 ;
print longest row number in Array Source program/L= 31
======== Print The duplicate source program ======
1. if    (    XXX    < Y ) // Y is ID
2.     X  = XXX     + 100 ;
Print The duplicate source program without comment one line
1. if    (    XXX    < Y )
2.     X  = XXX     + 100 ;
Print source program compress consecutive whitespaces into one.
1. if ( XXX < Y )
2.  X = XXX + 100 ;
========================================================
Lexs_oneLine [0]=if ( XXX < Y )
Lex_one[0]=if
Lex_one[1]=(
Lex_one[2]=XXX
Lex_one[3]=<
Lex_one[4]=Y
Lex_one[5]=)
========================================================
Lexs_oneLine [1]= X = XXX + 100 ;
Lex_one[6]=X
Lex_one[7]==
Lex_one[8]=XXX
Lex_one[9]=+
Lex_one[10]=100
Lex_one[11]=;
=======Print lexemes table======
1. if
2. (
3. XXX
4. <
5. Y
6. )
7. X
8. =
9. XXX
10. +
11. 100
12. ;
============== The End ==================
```

لاحظ: تم معالجة الفراغات المتتالية لكن لم يتم معالجة الاسطر الفارغة والفراغ اذا جاء في بداية أو نهاية الجملة في حالة كان المثال يحتوي على اسطر فارغة أو فراغات في بداية السطر او نهايته.

لاحظ في هذه الطريقة:
تم استخدام Lexs_oneLine وهي مصفوفة أحادية الابعاد موقته وضيفتها خزن صف واحد فقط يؤخذ من مصفوفة برنامج المصدر المعدلة G في كل مرة لدوارة i .
كما تم استخدام Lex_one وهي مصفوفة أحادية الابعاد موقته يخزن بداخلها Lexeme واحد فقط يؤخذ من مصفوفة Lexs_oneLine ثم يتم نقل محتوى هذه المصفوفة Lex_one ويوضع في صف مستقل لمصفوفة Lexs_table .

(4)

Source program G	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]	[15]	…	…	…	…	[99]
[0]	i	f		(		X	X	X		<		y		)							\0
[1]	X		=		X	X	X		+		1	0	0		;						\0

Lexs_oneLine
Loop i = 0

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]
i	f		(		X	X	X		<		y		)	\0

Lex_one
Loop k = 0 , 1 , 2
Counter M = 0 , 1 , 2

[0]	[1]	[2]
i	f	\0

Lex_one
Loop k = 3 , 4
Counter M = 0 , 1

[0]	[1]
(	\0

Lex_one
Loop k = 5 , 6 , 7 ,8
Counter M=0 , 1 , 2 , 3

[0]	[1]	[2]	[3]
X	X	X	\0

Lex_one
Loop k = 9 , 10
Counter M = 0 , 1

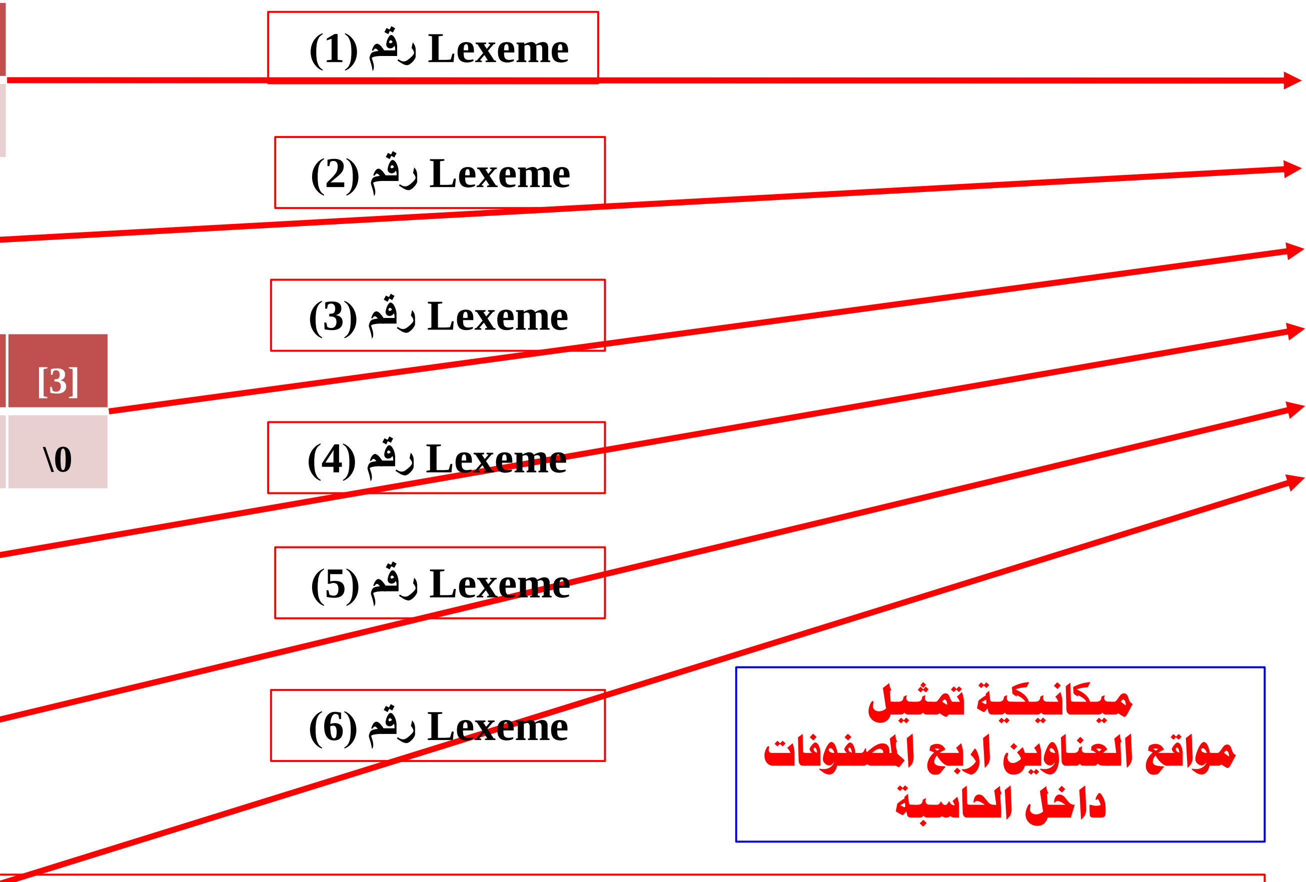
[0]	[1]
<	\0

Lex_one
Loop k = 11 , 12
Counter M = 0 , 1

[0]	[1]
y	\0

Lex_one
Loop k = 13 , 14
Counter M = 0 , 1

[0]	[1]
)	\0



(1) رقم Lexeme

(2) رقم Lexeme

(3) رقم Lexeme

(4) رقم Lexeme

(5) رقم Lexeme

(6) رقم Lexeme

ميكانيكية تمثيل
مواقع العناوين اربع المصفوفات
داخل الحاسبة

Lexs_table
Counter n = 0,1,2,3,4,5

	[0]	[1]	[2]	[3]
[0]	i	f	\0	
[1]	(	\0		
[2]	X	X	X	\0
[3]	<	\0		
[4]	Y	\0		
[5]	)	\0		
[6]				
[7]				
[8]				
[9]				
[10]				
[11]				

A virtual representation of address locations in four arrays, counters, and the number of Lexemes inside the computer (on paper).

(5)

Source program G	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]	[15]	…	…	…	…	[99]
[0]	i	f		(		X	X	X		<		y		)							\0
[1]	X		=		X	X	X		+		1	0	0		;						\0

Lexs_oneLine Loop i = 1	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]	[15]
	X		=		X	X	X		+		1	0	0		;	\0

Lexs_table
Counter n =
0,1,2,3,4,5,6,7,8,9,10,11

A virtual representation of address locations in four arrays, counters, and the number of Lexemes inside the computer (on paper).

ميكانيكية تمثيل
مواقع العناوين اربع المصفوفات
داخل الحاسبة

Lex_one
Loop k = 0 , 1
Counter M = 0 , 1

[0]	[1]
X	\0

(7) رقم Lexeme

Lex_one
Loop k = 2 , 3
Counter M = 0 , 1

[0]	[1]
=	\0

(8) رقم Lexeme

Lex_one
Loop k = 4 , 5 , 6 , 7
Counter M=0 , 1 , 2 , 3

[0]	[1]	[2]	[3]
X	X	X	\0

(9) رقم Lexeme

Lex_one
Loop k = 8 , 9
Counter M = 0 , 1

[0]	[1]
+	\0

(10) رقم Lexeme

Lex_one
Loop k = 10,11,12,13
Counter M=0 , 1 , 2 , 3

[0]	[1]	[2]	[3]
1	0	0	\0

(11) رقم Lexeme

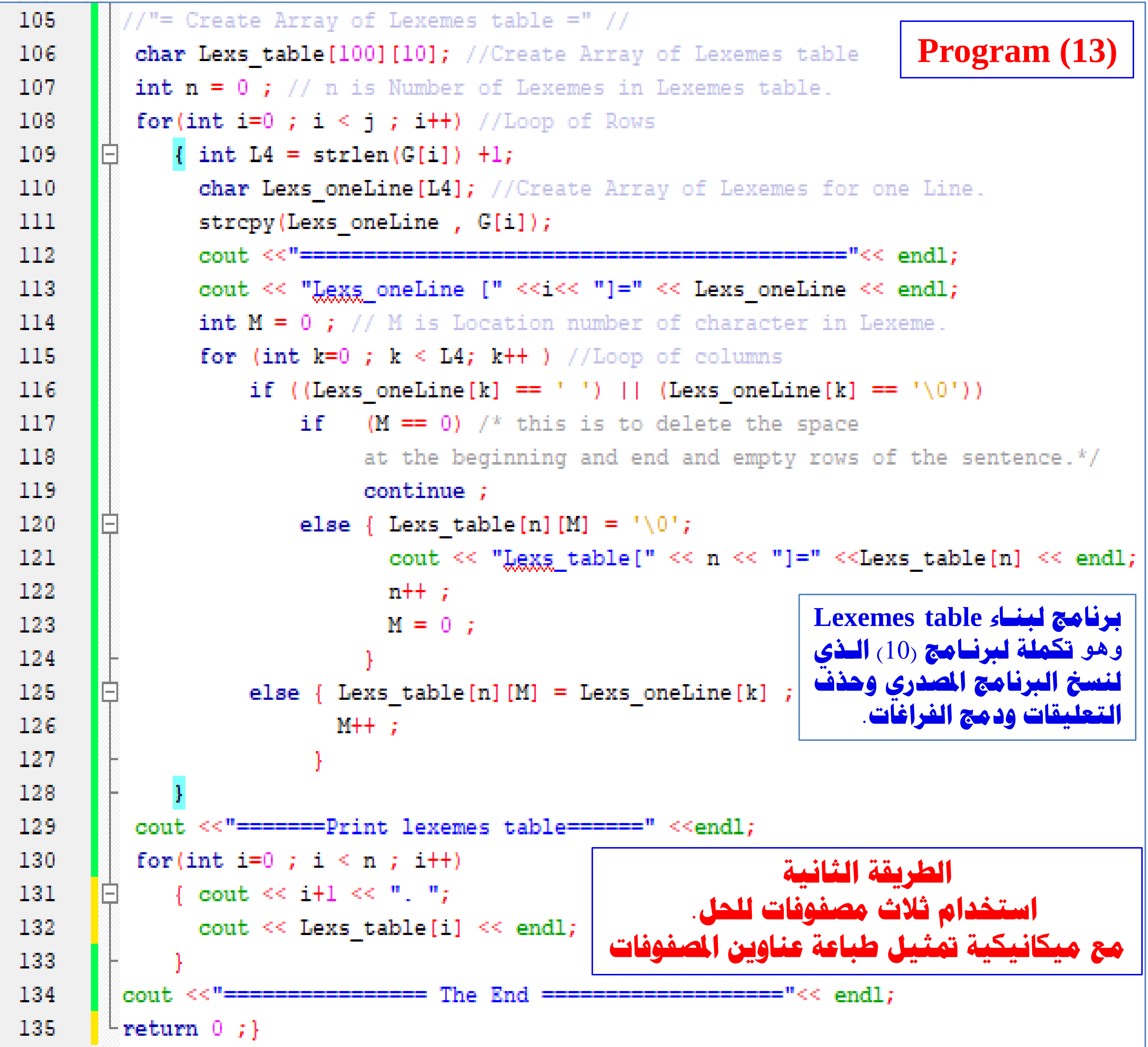
Lex_one
Loop k = 14 , 15
Counter M = 0 , 1

[0]	[1]
;	\0

(12) رقم Lexeme

	[0]	[1]	[2]	[3]
[0]	i	f	\0	
[1]	(	\0		
[2]	X	X	X	\0
[3]	<	\0		
[4]	Y	\0		
[5]	)	\0		
[6]	X	\0		
[7]	=	\0		
[8]	X	X	X	\0
[9]	+	\0		
[10]	1	0	0	\0
[11]	;	\0		

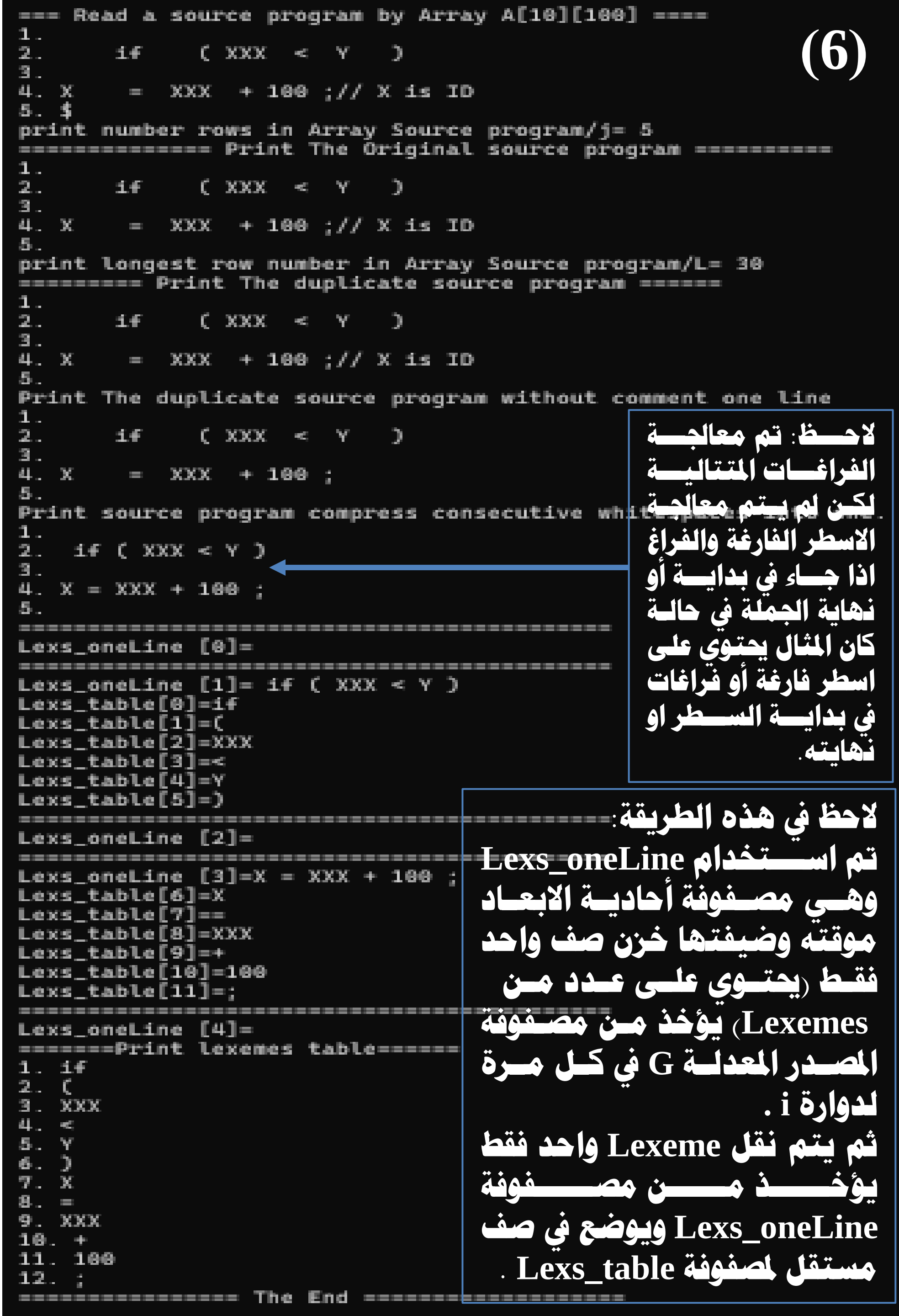
Program (13)



```
//"= Create Array of Lexemes table =" //
 char Lexs_table[100][10]; //Create Array of Lexemes table
 int n = 0 ; // n is Number of Lexemes in Lexemes table.
 for(int i=0 ; i < j ; i++) //Loop of Rows
    { int L4 = strlen(G[i]) +1;
      char Lexs_oneLine[L4]; //Create Array of Lexemes for one Line.
      strcpy(Lexs_oneLine , G[i]);
      cout <<"===========================================" << endl;
      cout << "Lexs_oneLine [" <<i<< "]=" << Lexs_oneLine << endl;
      int M = 0 ; // M is Location number of character in Lexeme.
      for (int k=0 ; k < L4; k++ ) //Loop of columns
          if ((Lexs_oneLine[k] == ' ') || (Lexs_oneLine[k] == '\0'))
              if   (M == 0) /* this is to delete the space
                     at the beginning and end and empty rows of the sentence.*/
                     continue ;
              else { Lexs_table[n][M] = '\0';
                     cout << "Lexs_table[" << n << "]=" <<Lexs_table[n] << endl;
                     n++ ;
                     M = 0 ;
                   }
          else { Lexs_table[n][M] = Lexs_oneLine[k] ;
                 M++ ;
               }
    }
 cout <<"=======Print lexemes table======" <<endl;
 for(int i=0 ; i < n ; i++)
    { cout << i+1 << ". ";
      cout << Lexs_table[i] << endl;
    }
cout <<"================ The End ===================" << endl;
return 0 ;}
```

برنامج لبناء Lexemes table وهو تكملة لبرنامج (10) الذي لنسخ البرنامج المصدري وحذف التعليقات ودمج الفراغات.

الطريقة الثانية
استخدام ثلاث مصفوفات للحل.
مع ميكانيكية تمثيل طباعة عناوين المصفوفات

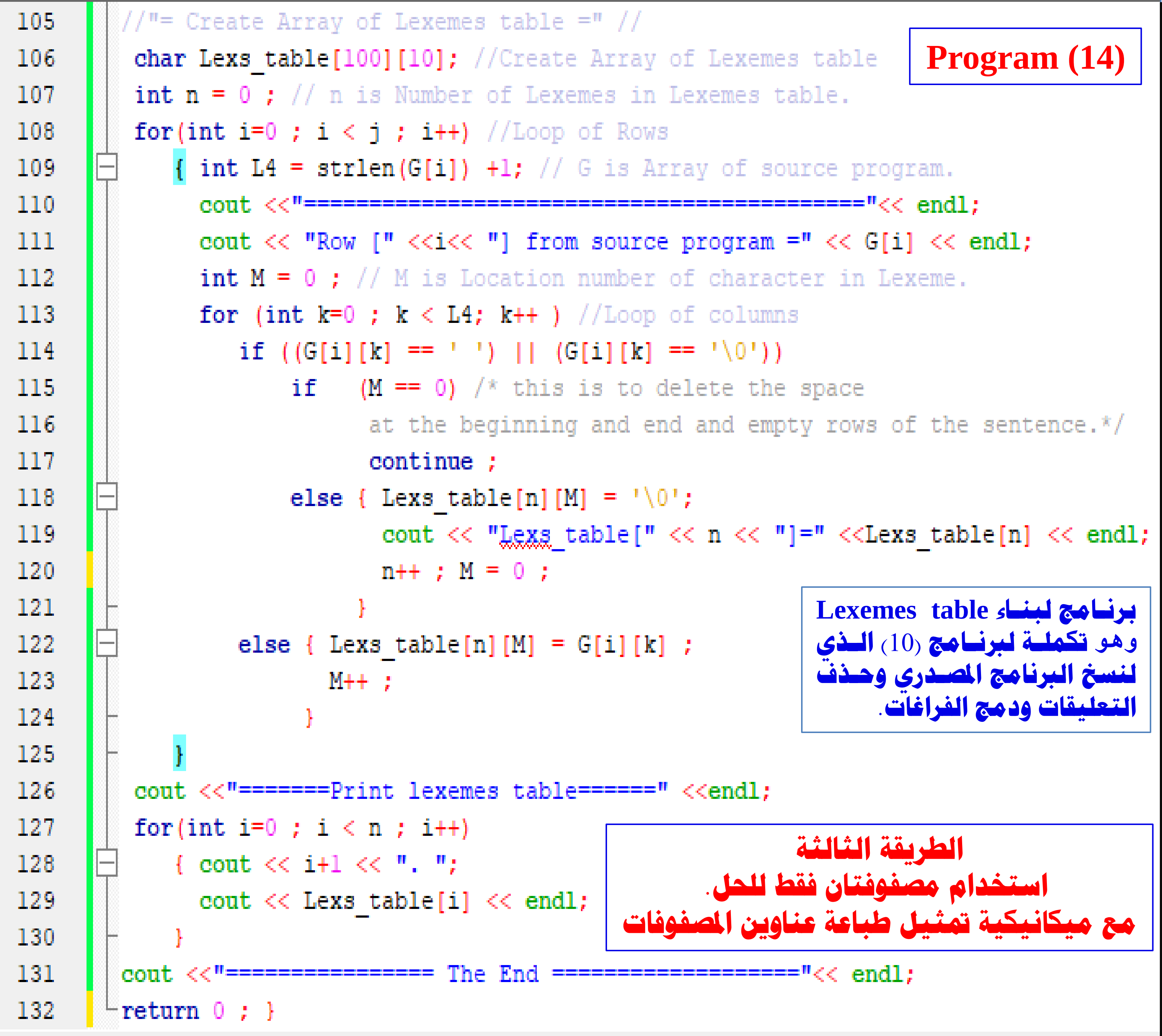


```
=== Read a source program by Array A[10][100] ====
1.
2.      if     ( XXX   <   Y    )
3.
4. X      =   XXX   + 100 ;// X is ID
5. $
print number rows in Array Source program/j= 5
============== Print The Original source program ==========
1.
2.      if     ( XXX   <   Y    )
3.
4. X      =   XXX   + 100 ;// X is ID
5.
print longest row number in Array Source program/L= 30
========= Print The duplicate source program ======
1.
2.      if     ( XXX   <   Y    )
3.
4. X      =   XXX   + 100 ;// X is ID
5.
Print The duplicate source program without comment one line
1.
2.      if     ( XXX   <   Y    )
3.
4. X      =   XXX   + 100 ;
5.
Print source program compress consecutive whitespace into one.
1.
2.   if ( XXX < Y )
3.
4. X = XXX + 100 ;
5.
=================================================
Lexs_oneLine [0]=
=================================================
Lexs_oneLine [1]= if ( XXX < Y )
Lexs_table[0]=if
Lexs_table[1]=(
Lexs_table[2]=XXX
Lexs_table[3]=<
Lexs_table[4]=Y
Lexs_table[5]=)
=================================================
Lexs_oneLine [2]=
=================================================
Lexs_oneLine [3]=X = XXX + 100 ;
Lexs_table[6]=X
Lexs_table[7]==
Lexs_table[8]=XXX
Lexs_table[9]=+
Lexs_table[10]=100
Lexs_table[11]=;
=================================================
Lexs_oneLine [4]=
=======Print lexemes table======
1. if
2. (
3. XXX
4. <
5. Y
6. )
7. X
8. =
9. XXX
10. +
11. 100
12. ;
================== The End ====================
```

لاحـظ: تم معالجة الفراغات المتتالية لكن لم يتم معالجة الاسطر الفارغة والفراغ اذا جاء في بداية أو نهاية الجملة في حالة كان المثال يحتوي على اسطر فارغة أو فراغات في بداية السطر او نهايته.

لاحظ في هذه الطريقة:
تم استخدام Lexs_oneLine وهي مصفوفة أحادية الابعاد موقته وضيفتها خزن صف واحد فقط (يحتوي على عدد من Lexemes) يؤخذ من مصفوفة المصدر المعدلة G في كل مرة لدوارة i .
ثم يتم نقل Lexeme واحد فقط يؤخذ من مصفوفة Lexs_oneLine ويوضع في صف مستقل لمصفوفة Lexs_table .

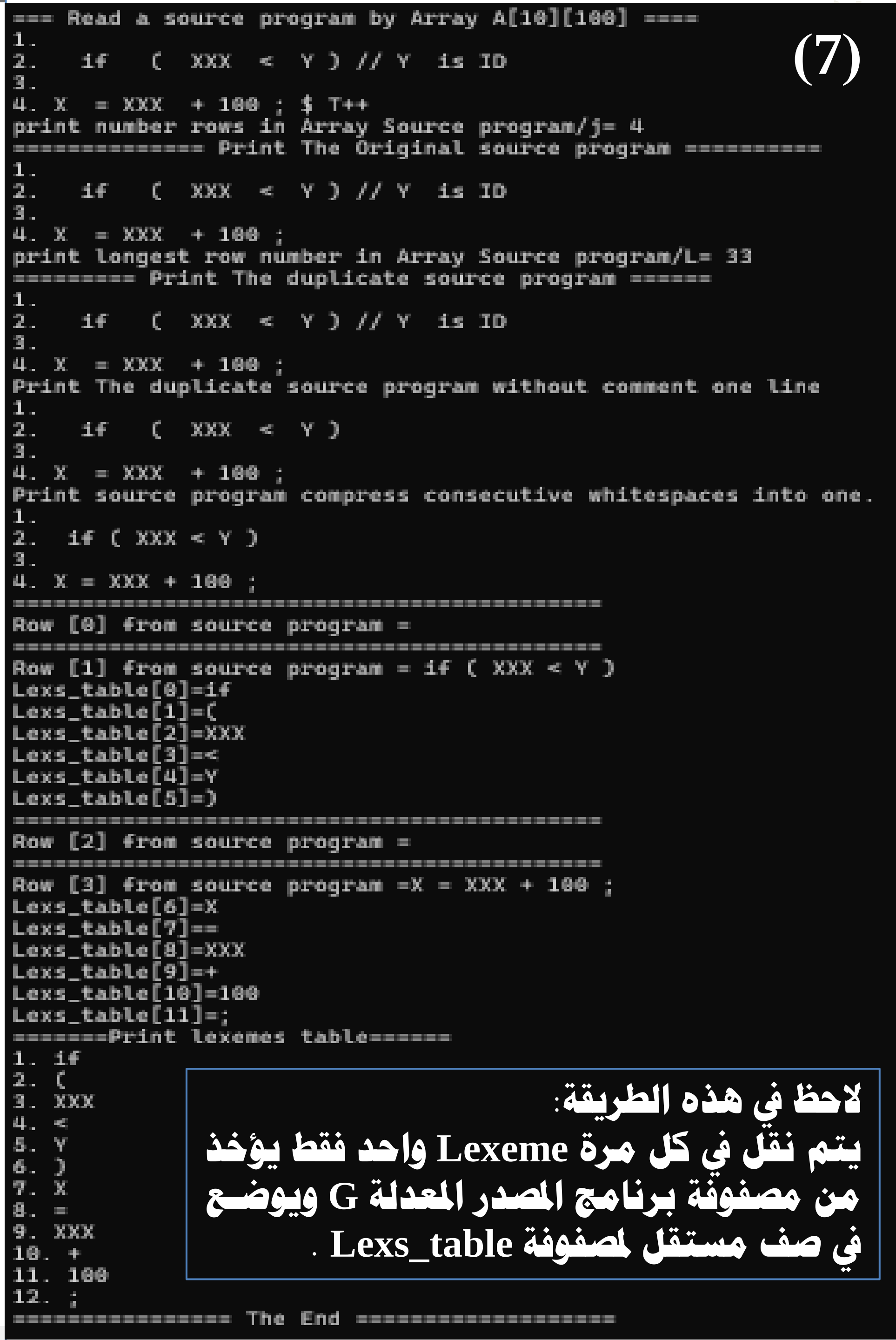
Program (14)



```
//"= Create Array of Lexemes table =" //
 char Lexs_table[100][10]; //Create Array of Lexemes table
 int n = 0 ; // n is Number of Lexemes in Lexemes table.
 for(int i=0 ; i < j ; i++) //Loop of Rows
    { int L4 = strlen(G[i]) +1; // G is Array of source program.
      cout <<"=========================================="<< endl;
      cout << "Row [" <<i<< "] from source program =" << G[i] << endl;
      int M = 0 ; // M is Location number of character in Lexeme.
      for (int k=0 ; k < L4; k++ ) //Loop of columns
         if ((G[i][k] == ' ') || (G[i][k] == '\0'))
            if   (M == 0) /* this is to delete the space
                  at the beginning and end and empty rows of the sentence.*/
                  continue ;
            else { Lexs_table[n][M] = '\0';
                   cout << "Lexs_table[" << n << "]=" <<Lexs_table[n] << endl;
                   n++ ; M = 0 ;
                 }
         else { Lexs_table[n][M] = G[i][k] ;
                M++ ;
              }
    }
 cout <<"=======Print lexemes table======" <<endl;
 for(int i=0 ; i < n ; i++)
    { cout << i+1 << ". ";
      cout << Lexs_table[i] << endl;
    }
cout <<"=============== The End =================="<< endl;
return 0 ; }
```

برنامج لبناء Lexemes table وهو تكملة لبرنامج (10) الذي لنسخ البرنامج المصدري وحذف التعليقات ودمج الفراغات.

الطريقة الثالثة
استخدام مصفوفتان فقط للحل.
مع ميكانيكية تمثيل طباعة عناوين المصفوفات



```
=== Read a source program by Array A[10][100] ====
1.
2.    if    (   XXX   <   Y ) // Y  is ID
3.
4. X   = XXX   + 100 ; $ T++
print number rows in Array Source program/j= 4
============== Print The Original source program ==========
1.
2.    if    (   XXX   <   Y ) // Y  is ID
3.
4. X   = XXX   + 100 ;
print longest row number in Array Source program/L= 33
========= Print The duplicate source program ======
1.
2.    if    (   XXX   <   Y ) // Y  is ID
3.
4. X   = XXX   + 100 ;
Print The duplicate source program without comment one line
1.
2.    if    (   XXX   <   Y )
3.
4. X   = XXX   + 100 ;
Print source program compress consecutive whitespaces into one.
1.
2.   if ( XXX < Y )
3.
4. X = XXX + 100 ;
===========================================
Row [0] from source program =
===========================================
Row [1] from source program = if ( XXX < Y )
Lexs_table[0]=if
Lexs_table[1]=(
Lexs_table[2]=XXX
Lexs_table[3]=<
Lexs_table[4]=Y
Lexs_table[5]=)
===========================================
Row [2] from source program =
===========================================
Row [3] from source program =X = XXX + 100 ;
Lexs_table[6]=X
Lexs_table[7]==
Lexs_table[8]=XXX
Lexs_table[9]=+
Lexs_table[10]=100
Lexs_table[11]=;
=======Print lexemes table======
1. if
2. (
3. XXX
4. <
5. Y
6. )
7. X
8. =
9. XXX
10. +
11. 100
12. ;
================ The End ==================
```

لاحظ في هذه الطريقة:
يتم نقل في كل مرة Lexeme واحد فقط يؤخذ من مصفوفة برنامج المصدر المعدلة G ويوضع في صف مستقل لمصفوفة Lexs_table .

الواجبات

H.W.(5)

How to draw a virtual representation of address locations in Three arrays, counters, and the number of Lexemes inside the computer (on paper).

H.W.(6)

How to draw a virtual representation of address locations in Two arrays, counters, and the number of Lexemes inside the computer (on paper).

THANK YOU