J.NAWROCKI, M. ANTCZAK, H. ĆWIEK, W. FROHMBERG, A. HOFFA, M. KIERZYNKA, S. WĄSIK

TRANSLATORS

ZAD. 1. Write a LEX program which omits in analyzed text all the Pascal identifiers other than: `begin end procedure`. An identifier is a sequence of letters, digits and underscores (_) which does not start with a digit.

Input:

```
procedurs: begin+ eno.
```

Output:

```
: begin+ .
```

ZAD. 2(*). Write a LEX program which omits in analyzed text all words containing vowels.

Input:

```
Ala mm kota.
```

Output:

```
 mm .
```

ZAD. 3(*). Write a LEX program which replaces each sign other than letter with a dot.

Input:

```
Ala ma kota 1111
Osy lubia miod 2222
3333
```

Output:

```
Ala.ma.kota.....
Osy.lubia.miod.....
....
```

ZAD. 4. Write a LEX program which sums all the values in each row of the input file.

Input:

```
5 6 7 3
3 4
2
```

Output:

```
5 6 7 3 Suma = 23
3 4 Suma = 7
2 Suma = 2
```

ZAD. 5(*). In the input file there are integers, some of them negative. Write a LEX program which rewrites the integers proceeding each of them with a sign (plus or minus) and gives the total number of integers in the input file.

Input:

```
-31 4 54 6 -4 -665
```

Output:

```
-31 +4 +54 +6 -4 -665
The file consists of 6 integers.
```

(*) gwiazdką oznaczone są zadania, które nie są realizowane na ćwiczeniach i są przeznaczone do wykonania jako zadania domowe.

ZAD. 6. Write a LEX program which checks if a sequence in the input file is of the form a^nb^n. Use BEGIN macro and two states.

Input:

```
aaabbb

aabbb
```

Output:

```
aaabbb T
 T
aabbb F
```

ZAD. 7(*). Write a LEX program which checks if a sequence in the input file contains alternating numbers 0 and 1. Use BEGIN macro and two states.

Input:

```
010101

10101
1100
```

Output:

```
010101 T
 T
10101 T
1100 F
```

ZAD. 8. Write a YACC program which checks if the input contains a string of the form 1+2*1+. If so, the program writes *"Syntax OK!!!"*, else *"Syntax Error"* message.

Correct examples:

111221
121
11

Incorrect examples:

1
21
12
12121

ZAD. 9(*). Write a YACC program which checks if the input contains a string of the form 1*2+3+1*. If so, the program writes *"Syntax OK!!!"*, else *"Syntax Error"* message.

Correct examples:

1231
22333
12333311

Incorrect examples:

121
131
1321
12131

ZAD. 10. Given the below presented LEX parser, write a YACC program which checks if the input file contains a sequence of digits, in which the digits are even and odd alternately. The sequence should start with an even digit.

(*) gwiazdką oznaczone są zadania, które nie są realizowane na ćwiczeniach i są przeznaczone do wykonania jako zadania domowe.

```
%{
  #include "ytab.h"
%}
%%

[02468]           {yylval=atoi(yytext);return p;}
[13579]           {yylval=atoi(yytext);return n;}
\n                ;
.                 {YY_FATAL("Unexpected character!!!");}
```

Correct examples:

21074

6

012

Incorrect examples:

1234

22

11

1

ZAD. 11. Given the parser from excercise 10, write a YACC program which for an input file consisting of a sequence of digits writes the sum of all odd digits.

Input:

```
017891
```

Output:

```
Sum: 18
```

ZAD. 12(*). Given the parser from excercise 10, write a YACC program which for an input file consisting of a sequence of digits writes if there are more even or odd digits.

Input:

```
017891
```

Output:

```
More odd digits.
```

ZAD. 13. Given the parser from excercise 10, write a YACC program which for an input file consisting of a sequence of digits rewrites the sequence adding brackets to it. Brackets should surround each subsequence consisting of odd numbers. You can assume that the first subsequence always consists of even numbers and the last of odd numbers.

Input:

```
01789134609
```

Output:

```
0(17)8(913)460(9)
```

ZAD. 14(*).Given the parser from excercise 10, write a YACC program which for an input file consisting of a sequence of digits rewrites the sequence adding brackets to it. Brackets should surround each subsequence consisting of odd numbers. You can assume that the first subsequence always consists of odd numbers and the last of even numbers.

.

Input:

```
17891346090
```

Output:

```
(17)8(913)460(9)0
```

(*) gwiazdką oznaczone są zadania, które nie są realizowane na ćwiczeniach i są przeznaczone do wykonania jako zadania domowe.