**علی نظری**

**9631075**

**نظریه زبان ها و ماشین ها**

**دکتر میبدی**

**Source Code:**

**import** java.util.ArrayList;  
**import** java.util.Scanner;  
  
**public class** Main  
{  
 **public static void** main(String[] args)  
 {  
 Scanner input = **new** Scanner(System.***in***);  
 System.***out***.print(**"Enter which Grammer you want?\n"** +  
 **"[1]. \n"** +  
 **"\tS->AB;\n"** +  
 **"\tA->aaa;\n"** +  
 **"\tB->aBb|?;\n"** +  
 **"[2]. \n"** +  
 **"\tS->aB|bB;\n"** +  
 **"\tB->aaB|abB|baB|bbB|?;\n"** +  
 **"[3]. \n"** +  
 **"\tS->aS|bS|?;\n"** +  
 **"[4]. \n"** +  
 **"\tS->aBC|aSBC;\n"** +  
 **"\tCB->AD;\n"** +  
 **"\tAD->BC;\n"** +  
 **"\taB->ab;\n"** +  
 **"\tbB->bb;\n"** +  
 **"\tbC->bc;\n"** +  
 **"\tcC->cc;\n"** +  
 **"[5]. \n"** +  
 **"\tS->U|V;\n"** +  
 **"\tU->TaU|TaT|UaT;\n"** +  
 **"\tV->TbV|TbT|VbT;\n"** +  
 **"\tT->aTbT|bTaT|?;\n"** +  
 **"[6]. \n"** +  
 **"\tS->aS|bA;\n"** +  
 **"\tA->cA|?;\n"** +  
 **"So... which one?"**);  
 ArrayList<String> rules = *selectGrammar*(Integer.*parseInt*(input.nextLine()));  
 **while** (**true**)  
 {  
 String thisInput = input.nextLine();  
 **if** (thisInput.isEmpty())  
 {  
 **break**;  
 }  
 **else if** (*isValid*(rules, **"S"**, thisInput))  
 {  
 System.***out***.println(**"Accept.َ"**);  
 }  
 **else** {  
 System.***out***.println(**"Fail!"**);  
 }  
 }  
 }

**private static** ArrayList<String> selectGrammar(**int** thisGrammar)  
 {  
 ArrayList<String> rules = **new** ArrayList<>();  
 **switch** (thisGrammar)  
 {  
 **case** 1:  
 rules.add(**"S->AB"**);  
 rules.add(**"A->aaa"**);  
 rules.add(**"B->aBb"**);  
 rules.add(**"B->?"**);  
 **break**;  
 **case** 2:  
 rules.add(**"S->aB"**);  
 rules.add(**"S->bB"**);  
 rules.add(**"B->aaB"**);  
 rules.add(**"B->abB"**);  
 rules.add(**"B->baB"**);  
 rules.add(**"B->bbB"**);  
 rules.add(**"B->?"**);  
 **break**;  
 **case** 3:  
 rules.add(**"S->aS"**);  
 rules.add(**"S->bS"**);  
 rules.add(**"S->?"**);  
 **break**;  
 **case** 4:  
 rules.add(**"S->aBC"**);  
 rules.add(**"S->aSBC"**);  
 rules.add(**"CB->AD"**);  
 rules.add(**"AD->BC"**);  
 rules.add(**"aB->ab"**);  
 rules.add(**"bB->bb"**);  
 rules.add(**"bC->bc"**);  
 rules.add(**"cC->cc"**);  
 **break**;  
 **case** 5:  
 rules.add(**"S->U"**);  
 rules.add(**"S->V"**);  
 rules.add(**"U->TaU"**);  
 rules.add(**"U->TaT"**);  
 rules.add(**"U->UaT"**);  
 rules.add(**"V->TbV"**);  
 rules.add(**"V->TbT"**);  
 rules.add(**"V->VbT"**);  
 rules.add(**"T->aTbT"**);  
 rules.add(**"T->bTaT"**);  
 rules.add(**"T->?"**);  
 **break**;  
 **case** 6:  
 rules.add(**"S->aS"**);  
 rules.add(**"S->bA"**);  
 rules.add(**"A->cA"**);  
 rules.add(**"A->?"**);  
 **break**;  
 }  
 **return** rules;  
 }

**public static boolean** isValid(ArrayList<String> rules, String word, String input)  
 {  
 **if** (word.equals(input))  
 {  
 **return true**;  
 }  
 **if** (*countLower*(word) > input.length())  
 {  
 **return false**;  
 }  
 **for** (**int** i = 0; i < rules.size(); i++)  
 {  
 String base = rules.get(i).substring(0, rules.get(i).indexOf(**"-"**));  
 String result =  
 rules.get(i).substring(rules.get(i).indexOf(**">"**) + 1);  
 **if** (result.equals(**"?"**))  
 {  
 result = **""**;  
 }  
 **if** (word.contains(base))  
 {  
 **if** (*isValid*(rules, word.replaceFirst(base, result), input))  
 {  
 **return true**;  
 }  
 }  
 }  
 **return false**;  
 }  
  
 **private static long** countLower(String input)  
 {  
 **return** input.chars().filter((s) -> Character.*isLowerCase*(s)).count();  
 }  
}

**Test cases:**

**Grammer 1:**

**S->AB;**

**A->aaa;**

**B->aBb|?;**

**aaa**

**Accept.**

**bb**

**Fail!**

**abba**

**Fail!**

**aab**

**Fail!**

**Grammer 2:**

**S->aB|bB;**

**B->aaB|abB|baB|bbB|?;**

**aaa**

**Accept.**

**aba**

**Accept.**

**abba**

**Fail!**

**ab**

**Fail!**

**Grammer 3:**

**S->aS|bS|?;**

**aaabbb**

**Accept.**

**bbb**

**Accept.**

**babb**

**Accept.**

**babbbbb**

**Accept.**

**Grammer 4:**

**S->aBC|aSBC;**

**CB->AD;**

**AD->BC;**

**aB->ab;**

**bB->bb;**

**bC->bc;**

**cC->cc;**

**bbbb**

**Fail!**

**baab**

**Fail!**

**bab**

**Fail!**

**aabb**

**Fail!**

**Grammer 5:**

**S->U|V;**

**U->TaU|TaT|UaT;**

**V->TbV|TbT|VbT;**

**T->aTbT|bTaT|?;**

**abba**

**Fail!**

**aaa**

**Accept.**

**abab**

**Fail!**

**baba**

**Fail!**

**Grammer 6:**

**S->aS|bA;**

**A->cA|?;**

**aaab**

**Accept.**

**aaaa**

**Fail!**

**abbba**

**Fail!**

**aaabbb**

**Fail!**