BLG 311E PROJECT PHASE 2

We have tried to implement the CYK algorithm to C++ program that reads grammar and words at files. For this purpuse, we have to understand the CYK algorithm. We studied the CYK algorith and started to implement it to C++.

First of all, we coded our code (main.cpp) at Visual Studio and that is why we added (_CRT_SECURE_NO_WARNINGS). Our prgoram should work on VS but it did not work Dev-C++ on my computer. (But it worked on my partners computer on Dev-C++).

Our program needs to CNF grammar like CYK. If grammar is not CNF form, program can be crash or something. Grammar and words on text file should be similar to pdf file example.

When programs run it prints NO – YES to the screen and it writes these NO – YES results.txt too. We cannot implement bonus part so program just writes YES or NO to text file.

Program has 3 functions (concat, search_string and combine) and main. One string matrix (gramer[MAX][MAX]) and a global int (numOfPro-> number of productions) defined at the begining of the code. At the begining of the run, it opens 3 text file (grammar.txt, words.txt with ifstream [in] and results.txt with ofstream [out] format). Then it uses while loop to read all of words and until the end of file (eof). Each loop it reads grammar again and again and then checks word can be or cannot be produce with that grammar. After that, program print the result (yes/no) on cmd screen and writes result to the results.txt file. After each word it checks eof and if it reachs the end program closes 3 files and return 0.

We used <fstream> and <string> libraries for reading/writing files and other string functions.

Program starts with opening 3 text file and it checks word file's eof and then checks grammar file's eof. After that, it reads grammar.txt file line by line (getline). For the first line (terminals) it reads line and assigns terminals to terminal char array without space, newline etc. characters and assigns the first character to the start string. Then for the second line it reads nonterminal characters and put them into the nonTerminal char array wtih the same method. For the third line it reads grammar and finds the "->" symbol and " " (first space symbol) and hold them array index at integers.Then it checks the left side of grammar (Exp. "S") for the how many times it used. If it is the first time that nonterminal used then it put that symbol to gramer[i][0]. If that nonterminal used before it did not put that symbol at matrix. After the nonterminal part prgoram reads line and takes the part that after the "->" symbol (right side) and then put them to gramer matrix with their nonterminal index i (gramer[i][1] or gramer[i][2]). It checked the nonterminal usage before that, and if nonterminal is the same as before it just puts right side of the grammar to gramer[i][2].

After the reading grammar part it reads words and assigns them to matrix. For this part porgram uses concat and combine functions. Concat function concatenates unique non-terminals (first and second string) and return them. Combine function creates every combination of variables from first and second string (Exp. 11, 22, 12, 21). While doing this this funciton uses search_string function for combined string can be created or not. This combine funciton returns the combined string. Search_string funciton returns a concatenated string of variables which can produce with sent string. It uses concat function for concatenating result string and grammar matrix.

For the final part program checks the top of the matrix. If the top of the matrix has the start symbol it prints YES and writes this to result file, if not it prints NO. After this part it returns at the begining of the program (which reading the grammar part etc.) until it reachs the end of grammar file. When it reachs the eof it closes files and finishes the program.

BONUS: We colud not implement bonus part that's why we did not use numbers on the grammar file, we just skipped that part of each line.

GROUP MEMBERS:
1.Baran Kaya   150130032
2.Kadir Enes Karslıoğlu   150130047