Retro Basic Project

ตัวโปรแกรมจะเริ่มจากรับชื่อไฟล์เข้ามา ซึ่งหากชื่อไฟล์ไม่ถูกต้อง โปรแกรมจะแสดงข้อความบอกผู้ใช้และให้ผู้ใช้ใส่ชื่อไฟล์ใหม่อีกครั้ง เมื่อใส่ชื่อไฟล์ถูกต้อง ตัวโปรแกรมจะมีขั้นตอนการแปลงภาษา ดังนี้

Scanner: เริ่มจากการนำ Input มาแยกเป็น Token โดยเก็บค่าไว้ใน List[x][y] โดย x คือจำนวนบรรทัด และ y คือจำนวน token ในแต่ละ line จากนั้นจึงนำ token มาเช็คว่าผิด syntax หรือไม่ โดยผ่านเงื่อนไข if else เช่นเช็คว่าตัวแรกของ line เป็น line_num หรือไม่ หรือตัวสุดท้ายเป็น operand หรือไม่ หากผิด โปรแกรมจะคืนค่า False และจบการทำงาน หากถูก โปรแกรมจะเข้าสู่ขั้นตอนต่อไป

การตรวจ Syntax หากตรงตามเงื่อนไขข้อใดข้อหนึ่งจะถือว่า SyntaxError และจบการทำงาน

1. Token ตัวแรกของแต่ละ line ไม่อยู่ใน line_num
2. Token ตัวถัดมาเป็น operand
3. พบ Token ใดๆ ที่ไม่ได้อยู่ใน id หรือ ['IF','GOTO','PRINT','STOP'] หรือ operand หรือไม่อยู่ในช่วง 0<=Token<=1000
4. Token คือ command ที่ไม่ใช่ STOP หรือ operand ที่อยู่ท้ายสุดของ line
5. Token คือ operand และ Token ตัวถัดมาไม่ใช่ id หรือ const

Parser: จะทำการอ่านชุดของ Token ที่ผ่านการเช็ค syntax มาเรียบร้อยแล้ว และ generate BCode ออกมาใส่ไฟล์ใหม่ ซึ่งในที่นี้จะใช้เงื่อนไข if-else เข้ามาช่วย โดยมีขั้นตอนดังนี้

1. output ตัวแรกของแต่ละ line เป็น 10 เสมอ
2. output ตัวที่สองของแต่ละ line เป็น line_num เสมอ
3. ถ้าเจอ IF ให้นำ '13 0 ' ไปต่อ output
4. ถ้าเจอ GOTO ให้นำ '14 ' ไปต่อ output
5. ถ้าเจอ PRINT ให้นำ '15 0 ' ไปต่อ output
6. ถ้าเจอ STOP ให้นำ '16 0 ' ไปต่อ output
7. ถ้าเจอ operand ให้นำ '17 ' ตามด้วย index ของ operand ไปต่อ output
8. ถ้าเจอ id ให้นำ '11 ' ตามด้วย index ของ id ไปต่อ output

   นอกเหนือจาก 8 ขั้นตอนนี้คือ constant โดยมีขั้นตอนการ parse ดังนี้

   1. ถ้า Token ก่อนหน้าเป็น id หรือ const แปลว่าต้อง GOTO ไป line นั้น ให้นำ '14 ' ตามด้วย Token นั้น ไปต่อ output
   2. ถ้า Token ก่อนหน้าคือ GOTO ให้นำ Token ตำแหน่งปัจจุบันไปต่อ output ได้เลย
   3. นอกเหนือจากกรณีที่กล่าวมา ให้นำ '12 ' ตามด้วย Token ตำแหน่งปัจจุบันไปต่อ output

Code (python)

```
id = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
const=range(101)
line_num=range(1001)
command = ['IF','GOTO','PRINT','STOP']
op = ['+','-','<','=']

def scanner(file):
    token_list = []
    for line in file:
        if line[:-1].split(' ')!=['']:
            token_list.append(line.strip(' ').replace('\n','').split(' '))
    for i in token_list:
        if i[0] not in line_num:
            return False
        if i[1] in op:
            return False
        for j in range(1,len(i)):
            if i[j] not in id:
                if i[j] not in command:
                    if i[j] not in op:
                        if i[j] not in line_num:
                            return False
            if j+1 == len(i):
                if i[j] in command or i[j] in op:
                    if i[j] != 'STOP':
                        return False
            if i[j] in op and i[j+1] not in id and i[j+1] not in const:
                return False
    return token_list
```

ในส่วนนีจะเก็บค่าคงทีต่างๆ

For อันแรกคือแยก line of file เป็น token

For ลูปใหญ่นีคือตรวจสอบ syntax ตามวิธีทีได้กล่าวไว้ข้างต้น

```
def parser_to_file(token):
    out_file = open("output"+filename,"w+")
    out = ''
    for i in token:
        out += '\n10 '
        for j in range(len(i)):
            if j == 0:
                out += str(i[j])+' '
            else:
                if i[j] == 'IF':
                    out += '13 0 '
                elif i[j] == 'GOTO':
                    out += '14 '
                elif i[j] == 'PRINT':
                    out += '15 0 '
                elif i[j] == 'STOP':
                    out += '16 0'
                elif i[j] in op:
                    out += '17 '+str(op.index(i[j])+1)+' '
                elif i[j] in id:
                    out += '11 '+str(id.find(i[j])+1)+' '
                else:
                    if j > 1 and i[j-1] not in command and i[j-1] not in op:
                        out += '14 '
                    elif j > 1 and i[j-1] == 'GOTO':
                        out += ''
                    else:
                        out += '12 '
                    out += str(i[j])+' '
    out += '\n0'
    out_file.write(out)
    out_file.close()
    print('')
    print(out)
    print('')
    print("finished write output")
```

ส่วนนีคือการ parse stream of token to BCode

```
while True:
    try:
        filename = input("input filename >> ").strip()
        file = open(filename, "r")
        if scanner(file) == False:
            print("InvalidSyntax")
            exit(0)
        break
    except FileNotFoundError:
        print("InvalidFilename")

parser_to_file(scanner(file))
file.close()
```

ส่วนนีคือส่วนของโปรแกรมหลักทีรับข้อมูลมาจากผู้ใช้

Link to my git
https://github.com/babe124578/Retro-basic

My raw file
https://raw.githubusercontent.com/babe124578/Retro-basic/master/retroParser.py

This PDF can found in my Github as my code