LAB 8 (RDP – Branching and Looping)

Bijay Regmi (210913032)

SOURCE CODE

```
print("Recursive Desent Parsing For C grammar\n")
#print("E->TE'\nE'->+TE'/@\nT->FT'\nT'->*FT'/@\nF->(E)/i\n")
#print("Enter the string want to be checked\n")
global s
program = input("Enter the program to be parsed : ")
s=list(program)
global i
i=0
def match(a):
    global s
    global i
    if(i>=len(s)):
        return False
    j = "".join(s[i:i+len(a)])
    if(j == a):
        print(f"{j} == {a} , Matched")
        i = i + len(a)
        return True
    else:
        return False


def Program():
    if(match("main(){")):
        if(declarations()):
            if(statement_list()):
                if(declarations()):
                    if(match("}")):
                        return True
                    else:
                        return False
                else:
                    return False
            else:
                return False
        else:
            return False
```

```
    else:
        return False


def declarations():
    if(data_type()):
        if(identifier_list()):
            if(match(";")):
                if(declarations()):
                    return True
                else:
                    return False
            else:
                return False
        else:
            return False
    else:
        return True


def data_type():
    if(match("int")):
        return True
    elif(match("char")):
        return True
    elif(match("float")):
        return True
    else:
        return False

def identifier_list():
    if(id()):
        if(X()):
            return True
        else:
            return False
    else:
        return False
def X():
    if(match(",")):
        if(identifier_list()):
            return True
        else:
            return False
    elif(match("[")):
        if(number()):
            if(match("]")):
                if(Y()):
```

```
                    return True
                else:
                    return False
            else:
                return False
        else:
            return False
    else:
        return True

def Y():
    if(match(",")):
        if(identifier_list()):
            return True
        else:
            return False
    else:
        return True


def statement_list():
    if(statement()):
        if(statement_list()):
            return True
        else:
            return False
    else:
        return True

def statement():
    if(assign_stat()):
        if(match(";")):
            return True
        else:
            return False
    elif(decision_stat()):
        return True
    elif(looping_stat()):
        return True
    else:
        return False

def assign_stat():
    if(id()):
        if(match("=")):
            if(expn()):
                return True
            else:
```

```
                return False
        else:
            return False
    else:
        return False

def expn():
    if(simple_expn()):
        if(eprime()):
            return True
        else:
            return False
    else:
        return False

def eprime():
    if(relop()):
        if(simple_expn()):
            return True
        else:
            return False
    else:
        return True

def simple_expn():
    if(term()):
        if(seprime()):
            return True
        else:
            return False
    else:
        return False

def seprime():
    if(addop()):
        if(term()):
            if(seprime()):
                return True
            else:
                return False
        else:
            return False
    else:
        return True

def term():
    if(factor()):
        if(tprime()):
```

```
            return True
        else:
            return False
    else:
        return False

def tprime():
    if(mulop()):
        if(factor()):
            if(tprime()):
                return True
            else:
                return False
        else:
            return False
    else:
        return True

def factor():
    if(id()):
        return True
    elif(number()):
        return True
    else:
        return False
def decision_stat():
    if(match("if(")):
        if(expn()):
            if(match("){")):
                if(statement_list()):
                    if(match("}")):
                        if(dprime()):
                            return True
                        else:
                            return False
                    else:
                        return False
                else:
                    return False
            else:
                return False
        else:
            return False
    else:
        return False

def dprime():
    if(match("else{")):
```

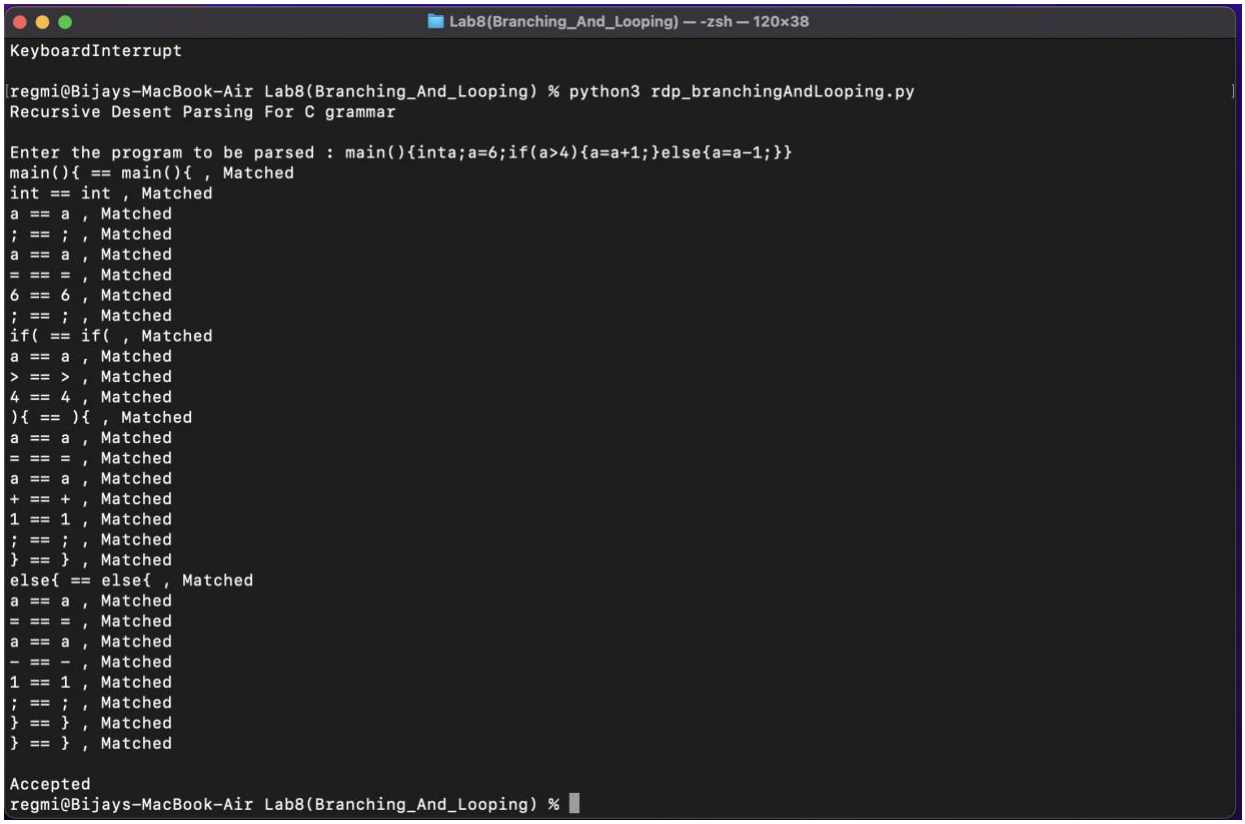
```
        if(statement_list()):
            if(match("}")):
                return True
            else:
                return False
        else:
            return False
    else:
        return True

def looping_stat():
    if(match("while(")):
        if(expn()):
            if(match("){")):
                if(statement_list()):
                    if(match("}")):
                        return True
                    else:
                        return False
                else:
                    return False
            else:
                return False
        else:
            return False
    elif(match("for(")):
        if(assign_stat()):
            if(match(";")):
                if(expn()):
                    if(match(";")):
                        if(assign_stat()):
                            if(match("){")):
                                if(statement_list()):
                                    if(match("}")):
                                        return True
                                    else:
                                        return False
                                else:
                                    return False
                            else:
                                return False
                        else:
                            return False
                    else:
                        return False
                else:
                    return False
            else:
```

```
                return False
        else:
            return False
    else:
        return False

def relop():
    if(match("==")):
        return True
    elif(match("!=")):
        return True
    elif(match(">")):
        return True
    elif(match("<")):
        return True
    else:
        return False
def addop():
    if(match("+")):
        return True
    elif(match("-")):
        return True
    else:
        return False

def mulop():
    if(match("*")):
        return True
    elif(match("/")):
        return True
    elif(match("%")):
        return True
    else:
        False

def id():
    if(match('a')):
        return True
    elif(match('b')):
        return True
    elif(match('c')):
        return True
    elif(match('d')):
        return True
    elif(match('e')):
        return True
    elif(match('j')):
        return True
```

```
    else:
        return False



def number():
    if(match('0')):
        return True
    elif(match('1')):
        return True
    elif(match('2')):
        return True
    elif(match('3')):
        return True
    elif(match('4')):
        return True
    elif(match('5')):
        return True
    elif(match('6')):
        return True
    elif(match('7')):
        return True
    elif(match('8')):
        return True
    elif(match('9')):
        return True
    else:
        return False

if(Program()):
    if(i == len(s)):
        print("\nAccepted")
    else:
        print("\nNot Accepted")
else:
    print("\nNot Accepted")
```

OUTPUT

1. BRANCHING



2. LOOPING