

Proiect LFT

Compiler Python

Student: Sopterean Andrei-Mihai

Grupa: 30235

1. Introducere

În acest proiect se dorește implementarea unui compilator al unui subset de instrucțiuni din limbajul Python. Proiectul va consta din 2 iterații, prima iterație este parsarea instrucțiunilor din subsetul ales și executia lor în C. A doua iterație a proiectului ar fi reprezentată de implementarea comportamentului unui compilator care ia un program sursă în limbajul Python scris folosind subsetul stabilit, îl parsează și construiește arbori de execuție din care să genereze cod de asamblare, echivalentul codului mașină.

2. Subset de instrucțiuni

- Variabile
- Atribuirii
- Operații aritmetice - +, -, *, /, %
- Condiții, operatori condiționali ==, !=, <, >, <=, >=
- Structuri de decizie – if
- Structuri repetitive – for, while

3. Tokenuri

- Variabila
- Operatori: +, -, *, /, %, ==, !=, <, >, <=, >=
- Condiție
- Instrucțiune
- Bloc de instrucțiuni
- if, for, while

4. Rulari

```
~/LFT/Proiect git P master ? cat input_files/debug_code_structure1.in
a = 10
if a:
    if a >= 10:
        if a < 10:
            b = 50
        if b > 10:
            b = 10
        else:
            b = 15
    if b != 15:
        if a == 10:
            c = 1
else:
    c = 0

~/LFT/Proiect git P master ? ./python_compiler < input_files/debug_code_structure1.in
ASSIGNATION
IF
    IF
        IF
            ASSIGNATION
        IF
            ASSIGNATION
        ELSE
            ASSIGNATION
    IF
        IF
            ASSIGNATION
ELSE
    ASSIGNATION
```

```
~/LFT/Proiect git P master ? ./python_compiler < input_files/debug_code_structure2.in
ASSIGNATION
IF
    IF
        IF
            ASSIGNATION
        ELSE
            ASSIGNATION
    ELSE
        IF
            ASSIGNATION
```

```
~/LFT/Proiect git P master ? cat input_files/debug_code_structure3.in
a = 10
if a:
    if a >= 10:
        if a < 10:
            b = 50
else:
    if a == 10:
        c = 1
```

```
~/LFT/Proiect git P master ? cat input_files/debug_code_structure3.in
a = 10
if a:
    if a >= 10:
        if a < 10:
            b = 50
else:
    if a == 10:
        c = 1

~/LFT/Proiect git P master ? ./python_compiler < input_files/debug_code_structure3.in
ASSIGNATION
IF
    IF
        IF
            ASSIGNATION
ELSE
    IF
        ASSIGNATION
```

```

~/LFT/Proiect git P master ? cat input_files/debug_code_execution1.in
n = 10
for i in range(10):
    for j in range(0, n):
        if i == j or i + j == n - 1:
            print("#")
        else:
            print("0")
    print("\n")

~/LFT/Proiect git P master ? ./python_compiler < input_files/debug_code_execution1.in
ASSIGNATION
FOR
    FOR
        IF
            PRINT
        ELSE
            PRINT
    PRINT
#00000000#
0#000000#0
00#0000#00
000#000#000
0000#0000
0000#0000
000#000#000
00#0000#00
0#000000#0
#00000000#

```

```

~/LFT/Proiect git P master ? cat input_files/debug_code_execution2.in
n = 16
print ("divizorii lui 16 sunt:")
print ("\n")
d = 2
while d < 16:
    if n % d == 0:
        print (d)
        print (" este divizor")
        print ("\n")
    else:
        print(d)
        print(" nu este divizor")
        print("\n")
    d = d + 1

~/LFT/Proiect git P master ? ./python_compiler < input_files/debug_code_execution2.in
ASSIGNATION
PRINT
PRINT
ASSIGNATION
WHILE
    IF
        PRINT
        PRINT
        PRINT
    ELSE
        PRINT
        PRINT
        PRINT
    ASSIGNATION
divizorii lui 16 sunt:
2 este divizor
3 nu este divizor
4 este divizor
5 nu este divizor
6 nu este divizor
7 nu este divizor
8 este divizor
9 nu este divizor
10 nu este divizor
11 nu este divizor
12 nu este divizor
13 nu este divizor
14 nu este divizor
15 nu este divizor

```

```
~/LFT/Proiect git P master ? cat input_files/debug_code_execution3.in
n = 25
print ("se calculeaza numere prime mai mici ca:")
print (n)
print ("\n")
nr = 2
while nr < n:
    prim = 1
    start = nr / 2
    for x in range(start, 1, -1):
        if nr % x == 0:
            prim = 0
    if prim:
        print(nr)
        print("\n")
    nr = nr + 1

~/LFT/Proiect git P master ? ./python_compiler < input_files/debug_code_execution3.in
ASSIGNATION
PRINT
PRINT
PRINT
ASSIGNATION
WHILE
    ASSIGNATION
    ASSIGNATION
    FOR
        IF
            ASSIGNATION
    IF
        PRINT
        PRINT
    ASSIGNATION
se calculeaza numere prime mai mici ca:25
2
3
5
7
11
13
17
19
23
```