CMPE150.02/03 2024 Fall Homework 3

TÜrkçe Programlama DİLim (TÜPDİL)

1. **Read all instructions very carefully**
2. **There might be updates in the homework description. Follow the Moodle announcements carefully.**
3. **Each update will be provided with a different color.**
4. **The grading is as follows. Otherwise no partial credit will be given in this homework (e.g. if the name of the submission is not correct)**

In this homework, you will implement a compiler and interpreter for the *TÜrkçe Programlama DİLim* (TÜPDİL). Your program should be able to handle a given program segment in TÜPDİL by performing one of the following actions:

1. Generate a compile error if there are syntax or semantic issues that prevent the code from being processed.
2. If no compile error, execute the statements. During execution:
    - Print out the requested values as specified in the program.
    - Complete the execution by either:
        - Finishing successfully, or
        - Ending with a runtime error if an issue arises during execution

Grading:

- 40: Only integers, no jump, no math operation.
- 50: Only integers, no jump, with math operations except bölü&çarp ~~without parentheses~~.
- 60: Only integers, with the jump, with math operations except bölü&çarp ~~without parentheses~~.
- 80: Only integers and floats, with the jump, with math operations ~~without parentheses~~.
- 100: All types, all statements.
- In all categories, >50% will test whether compile/runtime errors are correctly detected.

Name your solution file: your-student-number.py. Submit only this file without zip.

Read input from input.tup (reading from file code segment is provided). A non-empty input.tup will always be available.
Write output to file output.txt (writing to file code segment is provided).

Some example input.tup

```
Programı başlat.
A bir tam-sayı olsun.
A bir reel-sayı olsun.
A değeri 3,1 olsun.
```

```
Y bir tam-sayı olsun.
Y değeri 5 olsun.
Z bir reel-sayı olsun.
Z değeri 5 artı 4,1 çarp 6,2 olsun.
suç bir metin olsun.
Suç değeri !işte, bu! olsun.
Z çarp A eksi Y yazdır.
suç artı suç yazdır.
Programı bitir.
```

```
Programı başlat.
a bir reel-sayı olsun.
a değeri 3,1 olsun.
Y bir tam-sayı olsun.
Y değeri 5 olsun.
Z bir tam-sayı olsun.
Z değeri 5 artı ~~parantez-aç~~ 4 çarp 6 ~~parantez-kapa~~ olsun.
Z çarp Y eksi Y yazdır.
Programı bitir.
```

```
Programı başlat.
a bir reel-sayı olsun.
a değeri 3,1 olsun.
Y bir tam-sayı olsun.
Y değeri 7 olsun.
Y. satıra zıpla.
Z bir reel-sayı olsun.
Z değeri 3,1 çarp ~~parantez-aç~~ 4 çarp 6 ~~parantez-kapa~~ olsun.
Z değeri a çarp ~~parantez-aç~~ y çarp y ~~parantez-kapa~~ olsun.
Z çarp a eksi 3,2 yazdır.
Programı bitir.
```

You can assume that there will be no infinite loop.

There might be 6 different statements:

- Programı başlat.
- <var> bir <type> olsun.
- <var> değeri <expr> olsun.

- <expr> yazdır.
- <expr>. satıra zıpla.
- Programı bitir.

First parse the code line by line for a possible compile error. If there is a compile error (CE), provide the following error message:

Compile error at line <line-number>.

If there is no CE, start execution the program. If there is a runtime error (RTE), provide the following error message, execute the program until that error, print out if yazdır statement is executed before the RTE, and finally end the execution with the following error message:

Runtime error at line <line-number>.

Rules and Possible Compile Errors (CE) including syntax errors:

- The first and the last statements should be Programı başlat. and Programı bitir. If not, CE.
- ~~If a variable is assigned a value with an incorrect type, CE~~
    - ~~CE only if the expression in the assignment does not include any other variables.~~
- <var> should be composed of characters of a maximum length of 20, from the Turkish alphabet. If not, CE.
- <var> cannot take keywords of TÜPDİL such as
    - artı, eksi, çarp, bölü, metin. If not, CE
- <type> might be one of the following. If not, CE
    - tam-sayı
    - reel-sayı
    - metin
- <expr> might be one of the following. If not, CE
    - <var>
    - <value>
    - <exp> <op> <exp>
    - For simplicity, an <expr> will include either constant values or variables, i.e. an <expr> will not include both constants and variable.
    - ~~parantez-aç <exp> parantez-kapa~~
- <value> might be one of the following. In TÜPDİL, the maximum number of digits after the comma is 3 in TÜPDİL. The thousands separator is a dot and the decimal separator is a comma in Turkish. If these and the following rules are not followed, CE.
    - tam-sayı: An integer from -10.000 to +10.000. If a constant integer that exceeds these limits defined, CE.

    - reel-sayı: A float from -10.000,000 to +10.000,000. If a constant float that exceeds these limits defined, CE. If a constant float that has more than 3 digits after the comma, CE.
        - A constant reel-sayı should always include a decimal digit (when reading from the file). E.g. 3 is not a reel-sayı but 3,0 is a reel-sayı.
    - metin: A string, enclosed with exclamations (!), and is composed of only Turkish characters, numbers, spaces, commas, dots, colons, or semi-colons. Max number of characters is 50.
- <op> might be one of the following. If not, CE.
    - artı
    - eksi
    - çarp
    - bölü
- metin cannot be longer than 50 characters. If a constant metin value longer than 50 characters is defined, CE
- Each statement ends with a dot. If not, CE.
- Only a single space is used between items. Two or more successive spaces, tab character or newline character are not allowed. If not, CE.
- If a value is assigned to a variable that is not defined in the whole program, or an undefined variable is used in an expression CE. This CE can be produced as the last CE, after parsing all the statements.

More Rules and Possible Runtime Errors (RTE)

- Attempt to access a variable that has not been defined up to that point: RTE
- Attempt to access a variable whose value has not set up to that point: RTE
- If a variable is assigned a value with an incorrect type, RTE~~, only if the expression in the assignment DOES include at least one variable.~~

Math operations:

artı	tam-sayı	reel-sayı	metin
tam-sayı	sum up, result type: tam-sayı (RTE if exceeds the range)	CE/RTE	Convert metin to tam-sayı. sum up, result type: tam-sayı (RTE if metin is not an integer, RTE if exceeds the range)
reel-sayı	CE/RTE	sum up, result type: reel-sayı (RTE if exceeds the range)	Convert metin to reel-sayı. sum up, result type: reel-sayı (RTE if metin is not a float, RTE if exceeds the range, RTE if the decimal part exceeds 3 digits)

metin	Convert metin to tam-sayı. sum up, result type: tam-sayı (RTE if metin is not an integer, RTE if exceeds the range)	Convert metin to reel-sayı. sum up, result type: reel-sayı (RTE if metin is not a float, RTE if exceeds the range, RTE if the decimal part exceeds 3 digits)	Concat, result type: metin (RTE if exceeds the length)

eksi	tam-sayı	reel-sayı	metin
tam-sayı	subtract, result type: tam-sayı (RTE if exceeds the range)	CE/RTE	Convert metin to tam-sayı. subtract, result type: tam-sayı (RTE if metin is not an integer, RTE if exceeds the range)
reel-sayı	CE/RTE	subtract, result type: reel-sayı (RTE if exceeds the range)	Convert metin to reel-sayı. subtract, result type: reel-sayı (RTE if metin is not a float, RTE if exceeds the range, RTE if the decimal part exceeds 3 digits)
metin	Convert metin to tam-sayı. subtract, result type: tam-sayı (RTE if metin is not an integer, RTE if exceeds the range)	Convert metin to reel-sayı. subtract, result type: reel-sayı (RTE if metin is not a float, RTE if exceeds the range, RTE if the decimal part exceeds 3 digits)	Return a new metin, removing all instances of the right metin from the left metin

Example metin removals:

- !balalab! eksi !ala! returns !bb!
- !balabala! eksi !ala! returns !bb!
- !bbb! eksi !ala! returns !bbb!

çarp	tam-sayı	reel-sayı	metin
tam-sayı	çarp, result type: **reel-sayı** (RTE if exceeds the range)	CE/RTE	Return a new metin, repeating the original metin tam-sayı times (RTE if exceeds max

			length)
reel-sayı	CE/RTE	çarp, result type: reel-sayı (RTE if exceeds the range, RTE if the decimal part exceeds 3 digits)	Convert metin to reel-sayı. çarp, result type: reel-sayı (RTE if metin is not a float, RTE if exceeds the range, RTE if the decimal part exceeds 3 digits)
metin	Return a new metin, repeating the original metin tam-sayı times (RTE if exceeds max length)	Convert metin to reel-sayı. çarp, result type: reel-sayı (RTE if metin is not a float, RTE if exceeds the range, RTE if the decimal part exceeds 3 digits)	CE/RTE

bölü	tam-sayı	reel-sayı	metin
tam-sayı	böl, result type: reel-sayı (RTE if exceeds the range, RTE if the decimal part exceeds 3 digits)	CE/RTE	Convert metin to tam-sayı. böl, result type: reel-sayı (RTE if metin is not a float, RTE if exceeds the range, RTE if the decimal part exceeds 3 digits)
reel-sayı	CE/RTE	böl, result type: reel-sayı (RTE if exceeds the range, RTE if the decimal part exceeds 3 digits)	Convert metin to reel-sayı. böl, result type: reel-sayı (RTE if metin is not a float, RTE if exceeds the range, RTE if the decimal part exceeds 3 digits)
metin	Convert metin to tam-sayı. böl, result type: reel-sayı (RTE if metin is not a float, RTE if exceeds the range, RTE if the decimal part exceeds 3 digits)	Convert metin to reel-sayı. böl, result type: reel-sayı (RTE if metin is not a float, RTE if exceeds the range, RTE if the decimal part exceeds 3 digits)	CE/RTE

How to decide CE/RTE:

- If operation only depends on constant operands, or operators calculated from constant operands, this can be catched during compile time, therefore CE

- If operation includes variables, it will be catched during runtime, therefore RTE

Only the first CE is reported. If there is CE, no execution is done.

<expr>. satıra zıpla. statement:

- <expr> might evaluate tam-sayı, reel-sayı or metin.
- If <expr> is constant:
    - If cannot be converted to tam-sayı, CE.
    - If the corresponding line is out of range, CE.
- If <expr> is not constant:
    - If cannot be converted to tam-sayı, RTE.
    - If the corresponding line is out of range, RTE.

Math operations calculation:

- Inside parentheses are evaluated first. All math operations are evaluated from left to right.

metin cannot be longer than 50 characters. If a non-constant metin longer than 50 is created, RTE

Note that an already defined variable can be defined again with a new type. But it CANNOT take a value of a type different from its last definition.

<expr> yazdır. statement:

- The value is printed out with a newline.
- If <expr> is metin, it is printed out without exclamations.
- If <expr> is tam-sayı or reel-sayı, a dot should used for the thousands separator and the comma should used for the decimal separator.
- If <expr> is a reel-sayı and if the reel-sayı does not have a decimal digit, print a single decimal digit. E.g. for reel-sayı 34 print out 34,0