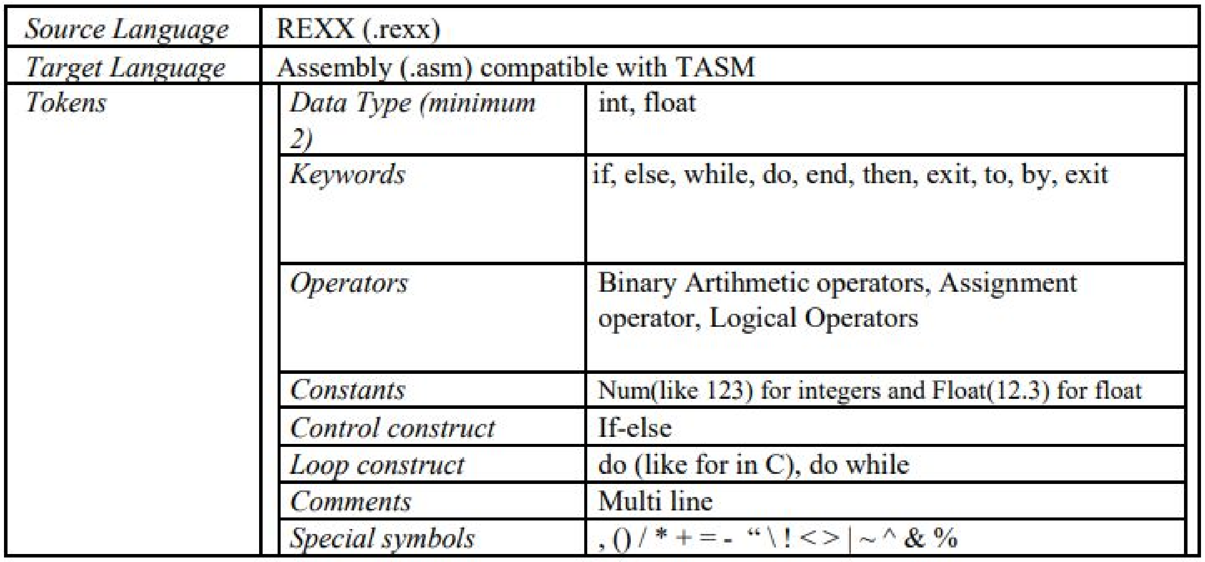
Innovative Assignment

17BCE049 Harsh Lunagariya

Rexx Language Compiler

Compiler Definition



Source Language	REXX (.rexx)	
Target Language	Assembly (.asm) compatible with TASM	
Tokens	*Data Type (minimum 2)*	int, float
	Keywords	if, else, while, do, end, then, exit, to, by, exit
	Operators	Binary Artihmetic operators, Assignment operator, Logical Operators
	Constants	Num(like 123) for integers and Float(12.3) for float
	Control construct	If-else
	Loop construct	do (like for in C), do while
	Comments	Multi line
	Special symbols	, () / * + = - “ \ ! < > | ~ ^ & %

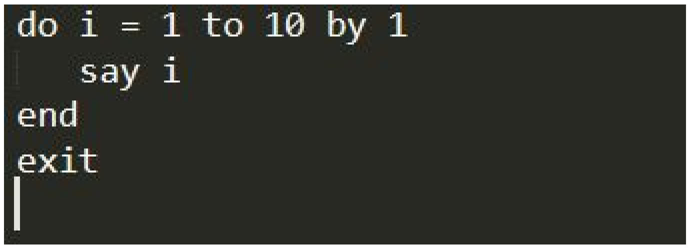
Compilation Steps

- Parse the token and other data using rexx.l file
- Checks the grammar using rexx.y file
- Generates AST and IR Tree using methods.h header
- Then traverse it and generate asm code using codeGen.h header

Example

Input/Output

Input 1

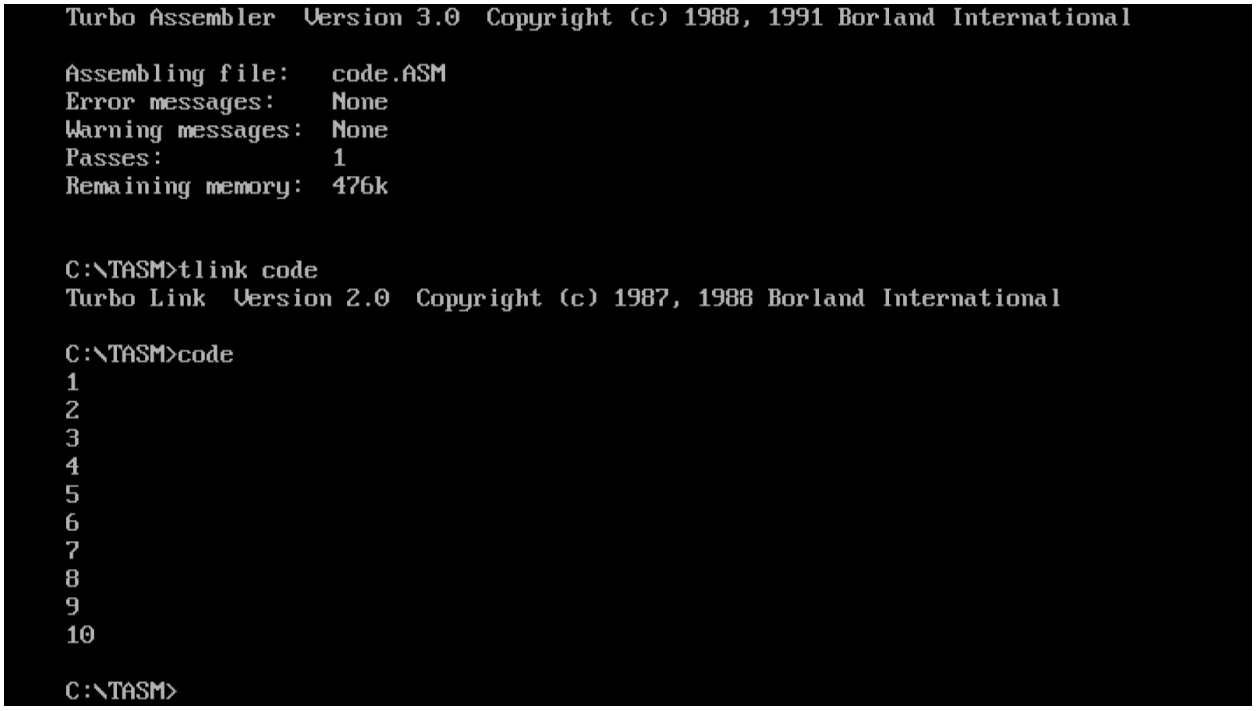


```
do i = 1 to 10 by 1
   say i
end
exit
```

Generated Code

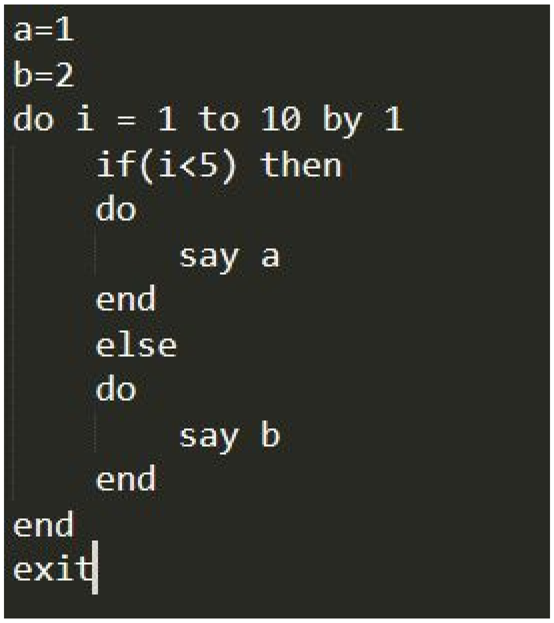
```
.model small
.stack 200h
.data

.code
mov ax , @data
mov ds , ax
mov ax , 0

mov BX, 1
push BX
pop BX
mov word ptr [ss:0],BX
L1:
mov BX, 10
push BX
pop BX
cmp BX,word ptr [ss:0]
jl L2
mov BX, [ss:0]
push BX
pop BX
mov AX, BX
call print
mov BX, 1
push BX
pop BX
add word ptr [ss:0],BX
jmp L1
L2:
mov ah,4ch
int 21h

print proc
    mov cx,0
    mov dx,0
    label1:
        cmp ax,0
        je print1
        mov bx,10
        div bx
        push dx
        inc cx
        xor dx,dx
        jmp label1
    print1:
        cmp cx,0
        je exit
        pop dx
        add dx,48
        mov ah,02h
        int 21h
        dec cx
        jmp print1
        exit:
    mov dx,13
      mov ah,2
      int 21h
      mov dx,10
      mov ah,2
      int 21h
        ret
print endp
end
```

Output



```
Turbo Assembler  Version 3.0  Copyright (c) 1988, 1991 Borland International

Assembling file:   code.ASM
Error messages:    None
Warning messages:  None
Passes:            1
Remaining memory:  476k


C:\TASM>tlink code
Turbo Link  Version 2.0  Copyright (c) 1987, 1988 Borland International

C:\TASM>code
1
2
3
4
5
6
7
8
9
10

C:\TASM>
```

Input 2



```
a=1
b=2
do i = 1 to 10 by 1
    if(i<5) then
    do
        say a
    end
    else
    do
        say b
    end
end
exit
```

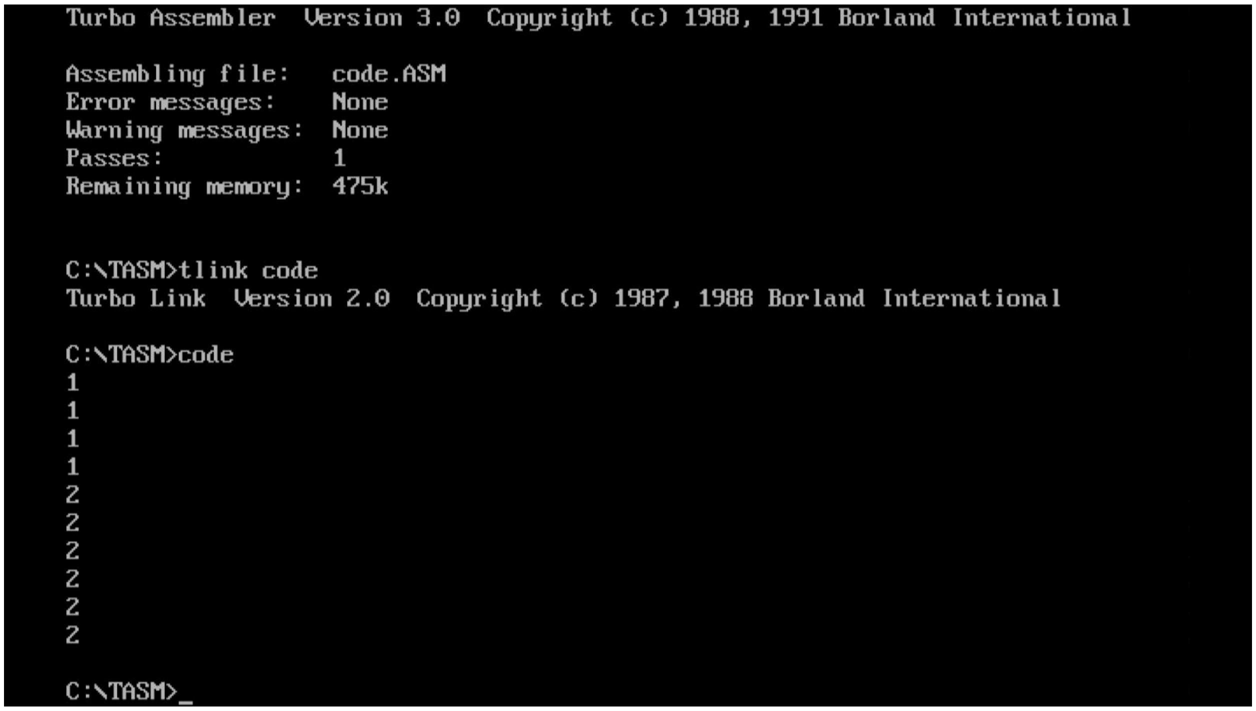
Generated Code

```
.model small
```

```
.stack 200h
.data

.code
mov ax , @data
mov ds , ax
mov ax , 0

mov BX, 1
push BX
pop BX
mov word ptr [ss:0],BX
mov BX, 2
push BX
pop BX
mov word ptr [ss:2],BX
mov BX, 1
push BX
pop BX
mov word ptr [ss:4],BX
L1:
mov BX, 10
push BX
pop BX
cmp BX,word ptr [ss:4]
jl L2
mov BX, 5
push BX
mov BX, [ss:4]
push BX
pop BX
mov AX,BX
pop BX
cmp AX, BX
jl L5
jnl L6
L5:
mov BX,1d
push BX
jmp L7
L6:
mov BX,0d
push BX
```

```
L7:
pop BX
cmp BX,0d
jg L3
mov BX, [ss:2]
push BX
pop BX
mov AX, BX
call print
jmp L4
L3:
mov BX, [ss:0]
push BX
pop BX
mov AX, BX
call print
L4:
mov BX, 1
push BX
pop BX
add word ptr [ss:4],BX
jmp L1
L2:
mov ah,4ch
int 21h

print proc
    mov cx,0
    mov dx,0
    label1:
        cmp ax,0
        je print1
        mov bx,10
        div bx
        push dx
        inc cx
        xor dx,dx
        jmp label1
    print1:
        cmp cx,0
        je exit
        pop dx
        add dx,48
        mov ah,02h
        int 21h
        dec cx
        jmp print1
        exit:
    mov dx,13
      mov ah,2
      int 21h
      mov dx,10
      mov ah,2
      int 21h
        ret
print endp
end
```

Output



```
Turbo Assembler  Version 3.0  Copyright (c) 1988, 1991 Borland International

Assembling file:   code.ASM
Error messages:    None
Warning messages:  None
Passes:            1
Remaining memory:  475k


C:\TASM>tlink code
Turbo Link  Version 2.0  Copyright (c) 1987, 1988 Borland International

C:\TASM>code
1
1
1
1
2
2
2
2
2
2

C:\TASM>_
```