MADE A PROGRAMMING LANGUAGE

Wanted to share it with u lot. Enjoy!!!

AHMED JALAL

TABLE OF CONTENTS

Syntax____3

Turing Completeness____ 4

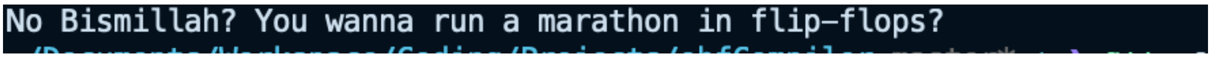
Obfuscation____5

Sample Programs____6

Syntax

Starting Point

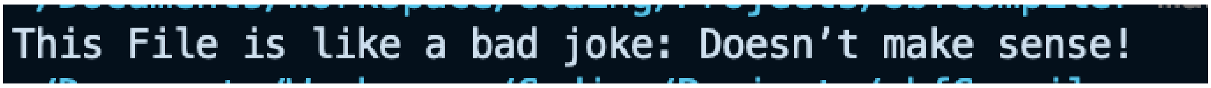
The Program’s starting point is “Bismillah” which should be on the first line of the program. Why? Cannot start yapping here… but yeah why not? On forgetting this, the compiler gives a sweet reminder.



```
No Bismillah? You wanna run a marathon in flip-flops?
```

Extension

The file containing the code has to have the extension “.bis”. Why? You may ask. I don’t really know. On incorrect file format, the compiler again passes a cute remark. And yeah it only corrects you on these 2 occasions, other than that you are on your own.

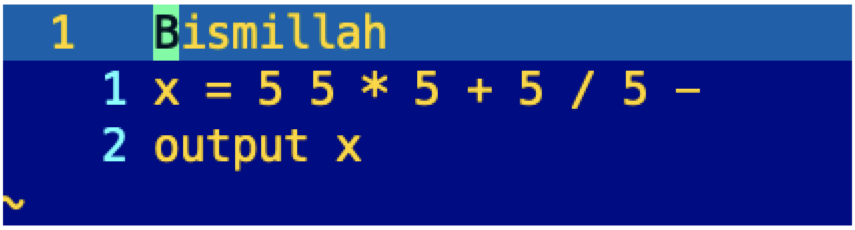


```
This File is like a bad joke: Doesn’t make sense!
```

Expression Evaluation

The expressions are evaluated in Reverse Polish Notation. It basically takes in the operands first and then the operator. For example, 5 5 + means 5 + 5. Also, the spaces between the operands, the operators and virtually anything in the whole language are very important. Take them as religiously as you take your semicolons. It might be a bit hard to chew on the first try but you’ll get used to it.

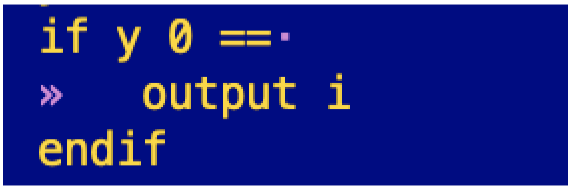
NOTE: Do not write complex expressions, as they might just evaluate wrong answers or break the compiler altogether(Sorry but I dont have the time to write a Parser). So restrain yourself to simple expressions.



```
Bismillah
x = 5 5 * 5 + 5 / 5 -
output x
```

Conditionals

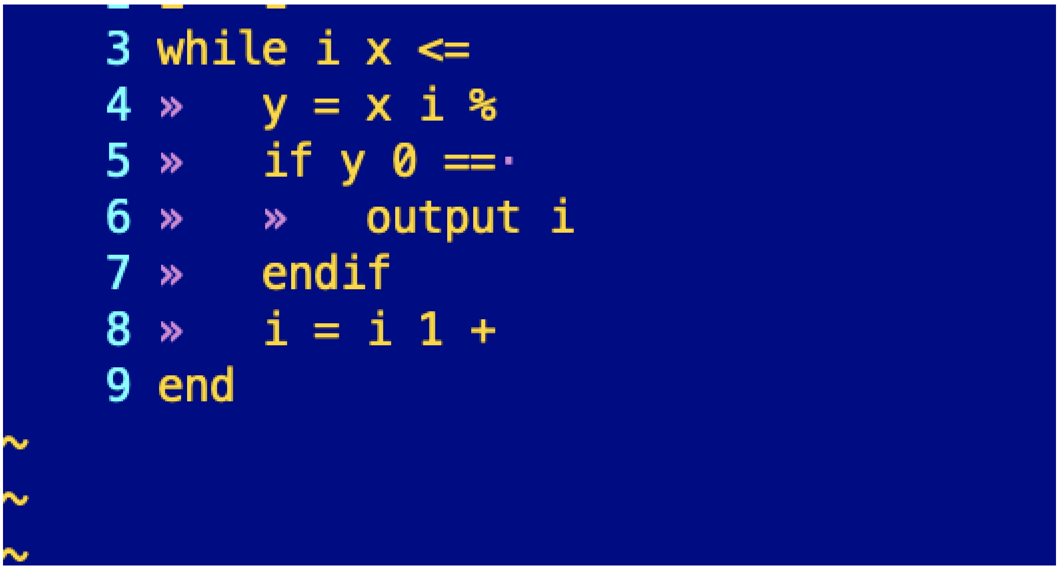
For now, only if conditional is implemented as all of the others can be emulated using it. The syntax is as following:



```
if y 0 ==
    output i
endif
```

Loops

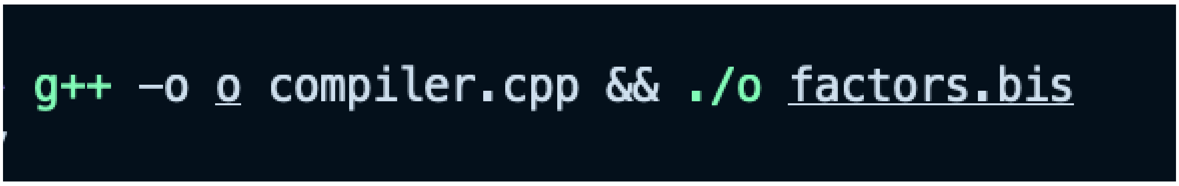
There is only one kind of loop in the language. This is due to the shortage of time and the fact that virtually every other loop can be built using it(in theory). The syntax is as following:



```
while i x <=
    y = x i %
    if y 0 ==
        output i
    endif
    i = i 1 +
end
```

Execution

To run your ".bis" file you just need to pass it as the first argument to the normal compilation of your program(compiler). "Compilation of a compiler", spooky isn't it? The command should look something like the following if you are using g++:



```
g++ -o o compiler.cpp && ./o factors.bis
```

Turing Completeness

In my humble opinion, this language is Turing complete but you can always try and prove me wrong. And this is one of the reasons for sharing this language with others. So that others can explore and build cooler stuff in it that I could not.

Limitations:

Following are some of the limitations of this language which are only there due to the shortage of time and we can work together on them if anyone is willing:

1. Cannot handle character data type
2. No clear distinction between integer and floating point values
3. Does not throw syntax or other errors (Don't have the time to build a parser)
4. No way to take input from users
5. Cannot handle complex expressions and fails in the DMAS heavy arena(Parser shi)
6. There are a lot more for sure that I cannot recall right now

NOTE: Most of the time the language is not going to give any syntax errors or anything. I am only a student and this language is just a personal project. So don't expect the compiler to behave like g++. Just ping me. I would know the problem as I have the deobfuscated code of the compiler.

Obfuscation

Sorry for obfuscating the code for the compiler. Just saving up for the final project. And please have some grace and not try to deobfuscate it. If you wanna do it anyway, do it I don't care. But if you wanna learn how this stuff works, just ping me. I promise it's easier than you might imagine.

Sample Programs

Some basic programs that I have written till now.

<—--->

Program to check if a number is palindrome or not:

```
Bismillah

question = 101
temp = question
answer = 0
numOfDigits = 0

while temp 0 >
      temp = temp 10 /
      numOfDigits = numOfDigits 1 +
end

i = 0
temp = question
while temp 0 >
      digit = temp 10 %
      answer = answer 10 * digit +
      i = i 1 +
      temp = temp 10 /
end

if answer question ==
      output 1
endif
if answer question !=
      output 0
endif
```

<—--->

Program to print factors of a number:

```
Bismillah
x = 20
i = 1
while i x <=
     y = x i %
     if y 0 ==
     output i
     endif
     i = i 1 +
end
```

<—-->

Program for a factorial:

```
Bismillah
x = 5
factorial = 1
while x 1 >=
     factorial = factorial x *
     x = x 1 -
end
output factorial
```

<—-->

Program to print Fibonacci series to a certain limit(terms):

```
Bismillah
terms = 20
a = 0
b = 1
i = 0
output a
output b
while i terms <
     fibo = a b +
     output fibo
     a = b
     b = fibo
     i = i 1 +
end
```

<—--->

Program for checking if a number is prime or not

```
Bismillah
x = 11
i = 2
bool = 1
while i x <
      y = x i %
      if y 0 ==
      bool = 0
      endif
      i = i 1 +
end
output bool
```

<—--->

Program for checking if a number is even or odd:

```
Bismillah
x = 5
y = x 2 %
if y 0 ==
      output 1
endif
if y 0 !=
      output 0
endif
```

<—--->

Program for printing the table of a number:

```
Bismillah
x = 5
i = 1
while i 10 <=
      output x i *
      i = i 1 +
end
```