Whitespace まだ門には入っていない

M.S.

2024 年 4 月 16 日

Whitespace はプログラミング言語の一つです。プログラミング言語というとアルファベットや記号が並んでいるのを想像するかと思いますが、ひらがなばかりの言語などもあり、Whitespace は見た目が**真っ白**なのが大きな特徴です。ずっと前から気になってはいたんですが、最近初めて仕組みを勉強したので、少しだけご紹介しようと思います。雰囲気だけでも感じ取ってもらえたらうれしいです。

私が Whitespace で初めて書いたプログラムは入力 n に対して $1+2+\cdots+n$ を表示するものでした。今回はもっとシンプルに $n+5$ を出力するプログラムを解説します。こんなんです。

return $n+5$

Python なら `print(int(input())+5)`

上のような見た感じ真っ白のコードにどのように命令が入っているのでしょうか？　Whitespace では、 （半角スペース）、 （Tab）、
（改行）の 3 種類の記号でプラグラムを書きます。分かりにくいので今後はそれぞれ `[SP]`、`[TAB]`、`[LF]` と書くことにします。私にとってはキー一押しでよかったのが「`¥texttt{[SP]}`」になって手間。。。

Whitespace では「スタック」と「ヒープ」というものを扱います。スタックは数字が書かれたブロックを机に積んでいくイメージで、例えば 0 を積む、とかします。ヒープはたくさん並んだ枠に 0 から始まる番号（アドレス）が振られていて、例えば 2 の枠（3 つめ！）に 5 という数字を書き込んだりします。簡単ですね。

では、次のような手順で入力 n に対して $n+5$ を出力しましょう。「[]」はスタックの土台、{ アドレス: 内容 } はヒープで、入力を n とします。

1. スタックに 0 を積む（0:[]）
2. スタックの一番上を複製する（0:0:[]）
3. スタックの一番上をアドレスとして取り出し、ヒープのその枠に入力された整数を書き込む（0:[]{0:n}）
4. スタックの一番上をアドレスとして取り出し、ヒープのその枠に書き込まれている整数をスタックに積む（n:[]{0:n}）
5. スタックに 5 を積む（5:n:[]{0:n}）
6. スタックの上 2 つを取り出し、その和を積む（$n+5$:[]{0:n}）
7. スタックの一番上を取り出し、整数として出力する（[]{0:n}）
8. プログラムを終了する

順に説明します。まず、スタックに整数を積むという 1 の操作は [SP][SP]<INT>です。<INT>は符号（正が [SP]、負が [TAB]）、絶対値（二進法、[SP]=0、[TAB]=1）、終わりを示す [LF] からなります。例えば 5 は二進法で 101 なので、5 を積む操作は [SP][SP][SP][TAB][SP][TAB][LF] です。ただ 0 を積むときは例外的に [SP][SP][SP][LF] でよいです。

次に、2 の操作は [SP][LF][SP] です。1 もそうだったように、スタックの操作はすべて [SP] で始まります。ほかにはスタックの上 2 つを交換する [SP][LF][TAB] などがあります。

3 の操作は [TAB][LF][TAB][TAB] です。入出力に関する操作は [TAB][LF] で始まります。最後の [TAB] を [SP] にすると入力の 1 文字目を文字として書き込みます。文字や記号には番号が振られており、正確にはその文字に対応する番号を書き込みます。

4 の操作は [TAB][TAB][TAB] です。ヒープへのアクセスは [TAB][TAB] で始まります。

5 の操作は 1 で説明しました。[SP][SP][SP][TAB][SP][TAB][LF] です。

6 の足し算は [TAB][SP][SP][SP] です。数値演算は [TAB][SP] で始まり、引き算（上から 2 番目 − 一番上）は [TAB][SP][SP][TAB]、掛け算は [TAB][SP][SP][LF]、割り算（上から 2 番目 ÷ 一番上、商は切り捨て）は [TAB][SP][TAB][SP] です。

7 は [TAB][LF][SP][TAB] です。入出力なので 3 同様 [TAB][LF] で始まります。最後の [TAB] を [SP] に変えると文字として出力します。

8 の終了は [LF][LF][LF] です。これ以降は読み込まれません。

以上を組み合わせると、次のようになります。

return $n+5$

```
[SP][SP][SP][LF]
[SP][LF]
[SP][TAB][LF]
[TAB][TAB][TAB][TAB][TAB][SP][SP][SP][TAB][SP][TAB][LF]
[TAB][SP][SP][SP][TAB][LF]
[SP][TAB][LF]
[LF]
[LF]
```

でもこれじゃあ Whitespace を書いたとは言えませんね。ちゃんと と と
で書くと、冒頭のようなプログラムになるのです。この 3 つ以外の文字や記号は書いても無視されます（ちゃんと動きます）。

気軽に試せる実行環境として、https://vii5ard.github.io/whitespace/ を紹介しておきます。デフォルトで書かれているコードを消して上のプログラムを書き、Run をクリックしてから下の枠に整数を入力し、Enter キーを押しましょう。5 足された数が出力されたでしょうか？　それができたら、上の説明を参考に $n-5, 5n, 3n+7, n^4, 3n^2+4n+2$ を出力するプログラムも書けるのではないでしょうか。入力された文字をそのまま出力すること

もできるはずです。プログラムを保存したい場合はメモ帳アプリでプログラムを作成して、このリンク先にコピー&ペーストするとよいでしょう。Whitespace には VSCode よりシンプルなメモ帳が向いています。

さらに詳しく知りたい、for 文や if 文も使いたいという方は、私が参考にした `https://susisu.github.io/wspace/ws.pdf` を読んでみてください。1 行目のタイトルから丁寧に読まないといつまでも入門できないと思いますが。