Building a Stack Machine

Note: 此題請用 yacc, lex 來實作

Description

在這題 BA 裡面你們需要以 yacc (bison), lex (flex) 來實作一個簡單的虛擬 stack machine 以處理此題的輸入。Stack machine 內部會執行 push, pop 與整數四則運算等操作，關於 stack machine 的規格會在下方說明，務必確認。

輸入為由 postfix notation 構成的 expression，例如 "3 4 −"，stack machine，最終編譯出來的可執行檔，在接收字串後會此輸入進行 tokenizing 以及 parsing，你需要做的是按照 stack machine 的規格在 .y source code 裡面實作一個 stack 以及相關的操作，每解析完一個 token 就執行 push 或加減乘除其中一種操作，其中加減乘除運算皆需執行兩次 pop ()

以 "3 4 −" 為例，stack machine 做的事情為

1. push 3 至堆疊上
2. push 4 至堆疊上
3. pop 兩次，第一次得到的值 4 作為減數，第二次 pop 的 3 則作為被減數。得到運算結果為 -1 (和 infix notation "3 - 4" 結果相同) 將此值 push 至堆疊上，其他運算依照同樣的方式計算，除法按照 C 語言裡 int 型別相除來作即可並且不需要考慮除以 0 的狀況

Input Format

每筆輸入為單行字串，字串表示由 postfix notation 構成的 expression，此 expression 不一定合法，你所實作的 stack machine 需要判斷不合法的原因為

1. stack 為空卻要執行 pop 或者是
2. stack 已滿卻要執行 push (只會有這兩種輸入不合法的情況)

並輸出相關執行時期的錯誤資訊並終止程式 (方法會在補充資訊提及)

Tokens Definition

- NUMBER ::= 0|[1-9][0-9]*|-[1-9][0-9]* (signed integer，不需考慮極大極小值)
- ADD ::= "+"
- SUB ::= "-”
- MUL ::= “*”
- DIV ::= “/” (不需考慮除以 0 的狀況)

Tokens 之間由單一空格作為間隔

Grammar

- startsymbol ::= expressions
- expressions ::= expression expressions | expression
- expression ::= ADD | SUB | MUL | DIV | NUMBER

Output Format

在每次 push 之後輸出 stack 內部所有元素，從 stack 底端開始直到頂端 (top)，元素之間以單一空格隔開，最後一個元素後面只需要輸出換行 (“\n”)，可直接參考下方 sample output

(output 中 "The contents of the stack are:" 冒號後方與 stack 最底端元素間也有一個空格)

在解析輸入的過程中遇到下列狀況時

1. stack 為空卻要執行 pop 或者是
2. stack 已滿卻要執行 push (只會有這兩種輸入不合法的情況)

分別輸出下列資訊

1. "Runtime Error: The pop will lead to stack underflow.\n"
2. “Runtime Error: The push will lead to stack overflow.\n”

輸出完 (注意一樣要換行) 終止程式 (方法會在補充資訊提及)

Spec of the Stack

```
%{
#include <stdio.h>
#define STACK_SIZE 10
int yylex();
void yyerror(const char* message);

struct stack {
    int data[STACK_SIZE];
    int top;
};
typedef struct stack stack_t;
stack_t stack;

int isEmpty(); // to check if the stack is empty
int isFull(); // to check if the stack is full
void push(int i);
int pop();
void dump(); // to dump (or print) all the data in stack
%}
```

之後記得在 `int main()` 裡面初始化 stack.top 的值

上方規格唯一有硬性規定一定要達成的只有 stack 的大小最多容納 10 個 signed integer，只要能夠達成 Output 的要求 functions 與堆疊不一定要照上方定義、實作

補充資訊

在解析輸入 expression 過程中如果發現輸入不合法，需要在輸出執行時期錯誤資訊後終止執行。相關的方法有好幾種而這裡需要使用的是 **YYABORT**，下方說明節錄自 gnu.org 的 bison manual

Macro: **YYABORT**

Return immediately with value 1 (to report failure).

需要注意此 macro 似乎只能在成對的 "%%" 之間 (文法定義區) 使用，使用前不需 include 其他標頭檔

下方為使用範例

```
%%

expression
   : NUMBER {
   // Some code
   YYABORT; // This will terminate the process
   // Some code
}  ;

%%
```

嘗試在自定義函式內使用會導致編譯錯誤 error: label ‘yyabortlab’ used but not defined，當然如果能夠解決找不到定義的問題那就能夠使用了

compiler-error

若遇到以下問題:

```
ba3.y:120:5: error: 'for' loop initial declarations are only allowed in C99 or C11 mode
    for(int i=0; i<=stack.top; i++){
    ^
ba3.y:120:5: note: use option -std=c99, -std=gnu99, -std=c11 or -std=gnu11 to compile your code
```

請在 for 迴圈 statement 上面先宣告 `int i;`

Sample Input	**Sample Output**
3 4 * 5 6 * +	The contents of the stack are: 3 The contents of the stack are: 3 4 The contents of the stack are: 12 The contents of the stack are: 12 5 The contents of the stack are: 12 5 6 The contents of the stack are: 12 30 The contents of the stack are: 42
1 2 3 * * /	The contents of the stack are: 1 The contents of the stack are: 1 2 The contents of the stack are: 1 2 3 The contents of the stack are: 1 6 The contents of the stack are: 6 Runtime Error: The pop will lead to stack underflow.
1 1 1 1 1 1 1 1 1 1 1	The contents of the stack are: 1 The contents of the stack are: 1 1 The contents of the stack are: 1 1 1 The contents of the stack are: 1 1 1 1 The contents of the stack are: 1 1 1 1 1 The contents of the stack are: 1 1 1 1 1 1 The contents of the stack are: 1 1 1 1 1 1 1 The contents of the stack are: 1 1 1 1 1 1 1 1 The contents of the stack are: 1 1 1 1 1 1 1 1 1 The contents of the stack are: 1 1 1 1 1 1 1 1 1 1 Runtime Error: The push will lead to stack overflow.
-1 0 4 - * 40 -100 10 / +	The contents of the stack are: -1 The contents of the stack are: -1 0 The contents of the stack are: -1 0 4 The contents of the stack are: -1 -4 The contents of the stack are: 4 The contents of the stack are: 4 40 The contents of the stack are: 4 40 -100 The contents of the stack are: 4 40 -100 10 The contents of the stack are: 4 40 -10 The contents of the stack are: 4 30