**HW - 1 Stack Machine**

1. 介紹

Stack Machine 是一種計算模型。記憶體以堆疊呈現。

簡單來說，堆疊機的運算是使用stack計算出postorder，且運算只對頂端生效。

以舉例：

Push 2 //stack：<2>

Push 3 //stack：<2,3>

MUL //stack：<6>

Push 4 //stack：<6,4>

Push 5 //stack：<6,4,5>

MUL //stack：<6,20>

ADD //stack：<26>

* Result：26

1. 操作說明
2. STEP1 建立執行檔

$ g++ -o main .\stackMachine.cpp

1. STEP2 開啟執行檔

開啟執行檔案，其參數為你的postorder字串

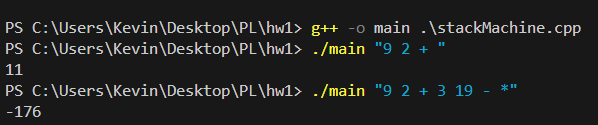
# argument is your postorder

$ ./main.exe "2 3 +"

5

postorder字串需要將每一個數字or運算元以空格分開，以避免發生混淆問題

1. 執行畫面參考

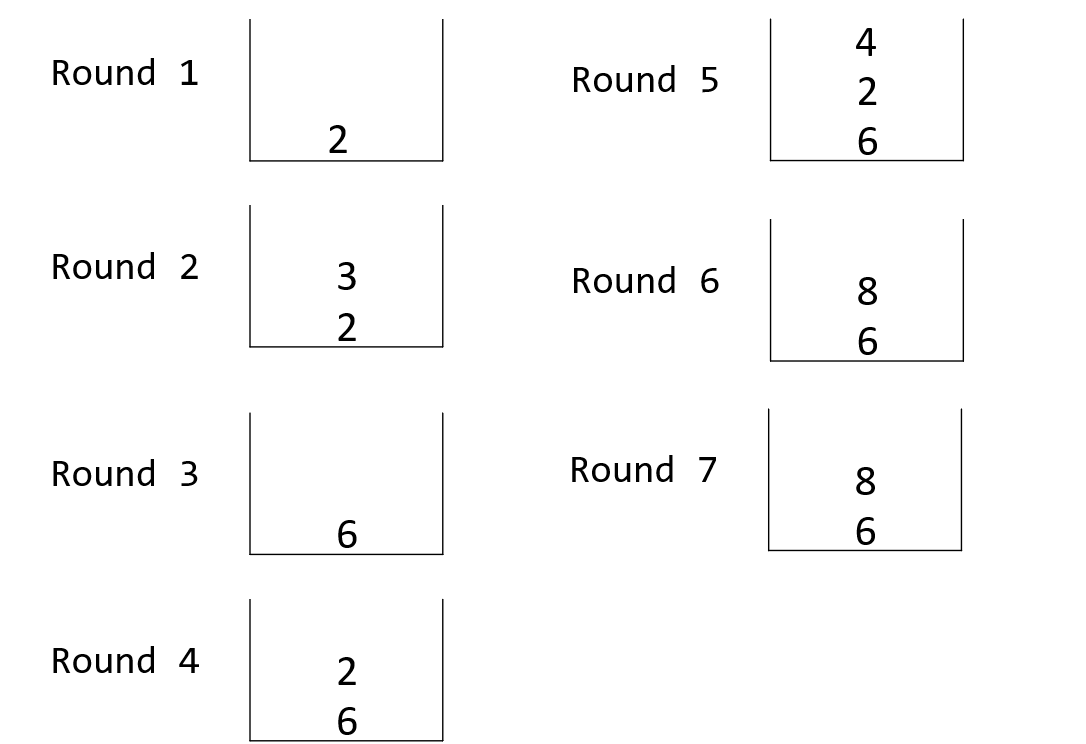


1. 程式碼
2. 完成程度：靠自己
3. 程式語言：C++

|  |
| --- |
| #include <iostream>  #include <stack>  using namespace std;  int main(int argc ,char \*\*argv){  if (argc!=2){  cerr<<"Use one Argument\n";  return 1;  }  // get argument  string express(argv[1]);  stack<int> st ;  int tmp = 0;  bool flag = false;  for (int i=0;i<express.size();){  if (!isdigit(express[i]) && express[i]!='+'  && express[i] != '-' && express[i] != '\*'  &&express[i] != '/' && express[i]!=' '  ){  cerr<<" Invalid Argument!\n";  return 1;  }  // space  if (express[i]==' '){  if (flag)  st.push(tmp);  i++;  flag = false;  tmp = 0;  continue;  }    if (isdigit(express[i])){  flag = true;  tmp = tmp\*10 + express[i] - '0';  i++;  continue;  }    if (express[i] == '+'){  if (st.size()<2){  cerr<<" Invalid Argument!\n";  return 1;  }  int n1 = st.top(); st.pop();  int n2 = st.top(); st.pop();  st.push(n2 + n1);  i++;  continue;  }  if (express[i] == '-'){  if (st.size()<2){  cerr<<" Invalid Argument!\n";  return 1;  }  int n1 = st.top(); st.pop();  int n2 = st.top(); st.pop();  st.push(n2 - n1);  i++;  continue;  }  if (express[i] == '\*'){  if (st.size()<2){  cerr<<" Invalid Argument!\n";  return 1;  }  int n1 = st.top(); st.pop();  int n2 = st.top(); st.pop();  st.push(n2 \* n1);  i++;  continue;  }  if (express[i] == '/'){  if (st.size()<2){  cerr<<" Invalid Argument!\n";  return 1;  }  int n1 = st.top(); st.pop();  int n2 = st.top(); st.pop();  if (n1 == 0){  cerr<<"Divide by zero\n";  return 1;  }  st.push(n2 / n1);  i++;  continue;  }  }  if (st.size()!=1){  cerr<<"Ivalid Argument!\n";  return 1;  }  cout<<st.top()<<endl;  st.pop();  return 0;  } |

1. 程式說明

* 我使用stack<int>st當成Stack Machine中的記憶體
* 如果遇到數字則直接push至st中，遇到operand將兩個元素pop出來並進行運算
* 以2 3 \* 2 4 \* +



1. 例外討論

* 本程式(執行檔)不接受超過一個參數



* 本程式在錯誤參數的情形如下
* Postorder 本身有誤：



* Divide by zero



* 無效的表達式





* + 1. Reference
    2. [低レイヤを知りたい人のためのCコンパイラ作成入門, 網路中譯版](https://koshizuow.gitbook.io/compilerbook/calculator_level_language/stack_machine/concept)