#include <stdio.h>

#include <stdlib.h>

#include <string.h>

typedef struct numStack {

int data;

struct numStack \*next;

}

numStack;

void push(numStack \*\*head, int numData) {

numStack\* node = malloc(sizeof(numStack));

node->data = numData;

node->next = \*head;

\*head = node;

}

int pop(numStack \*\*head) {

numStack\* top = \*head;

int value = top->data;

\*head = top->next;

free(top);

return value;

}

int numChecker(char\* mathNum) {

for(; \*mathNum; mathNum++)

if(\*mathNum < '0' || \*mathNum > '9') {

return 0;

}

return 1;

}

int mathSolver(char mathOpr, numStack\*\* top) {

int tempNum;

switch(mathOpr) {

case '+':

return pop(top) + pop(top);

case '-':

tempNum = pop(top);

return pop(top) - tempNum;

case '/':

tempNum = pop(top);

return pop(top) / tempNum;

case 'x':

return pop(top) \* pop(top);

}

}

int main(int argc, char\*\* argv) {

int stackSize = 0;

numStack\* topNode = NULL;

if(argc == 1){

printf("Input: %s\n", argv[0]);

exit(1);

}

for(int i = 1; i < argc; i++) {

char\* numCount = argv[i];

char mathOperator;

if(numChecker(numCount)) {

int temp = atoi(numCount);

push(&topNode, temp);

++stackSize;

}

else {

mathOperator = numCount[0];

push(&topNode, mathSolver(mathOperator, &topNode));

--stackSize;

}

}

printf("Result: %i\n", topNode->data);

return 0;

}