

問題用紙

2025	科目名	情報基礎（プログラミング）	1 / 3	通し番号	
------	-----	---------------	-------	------	--

逆ポーランド記法に基づいて実数の演算を行う計算機を C 言語を用いてプログラムリストのように作成する。演算に使用するスタックは、構造体 `node` をポインタで連結したリストとして表現する。外部変数（グローバル変数）`stack` は、スタックの最上部を指し、スタックが空の場合は `NULL` とする。図 1 はスタックの操作例、また、図 2 は `main` 関数の実行例である（図 1 は、図 2 の実行を図化したものではない。また、図 2 では、スタックの最上部が左に位置することに注意すること）。以下の問いに答えよ。

問 1 関数 `stack_depth` は、スタックの深さ（要素の個数に等しい）を返す関数である。空欄 A を埋めて関数 `stack_depth` を完成せよ。ただし、スタックが空の場合は、深さを 0 とする。

問 2 関数 `push` は、図 1 (1)、(2) に示すように、引数に与えられた実数をスタックの最上部に積む関数である。空欄 B を埋めて関数 `push` を完成せよ。ただし、メモリ不足等の例外は考慮しないものとする。

問 3 関数 `pop` は、図 1 (3) に示すように、スタックの最上部にある実数を取り出す（スタックは 1 段浅くなる）関数である。空欄 C1 と C2 を埋めて関数 `pop` を完成せよ。ただし、スタックが空の場合は、数値 0.0 を関数の値として返すものとする。また、関数 `free` は、関数 `malloc` で獲得したメモリを解放する関数である。

問 4 関数 `subtract` は、図 1 (4) に示すように、スタックの上から 2 番目の数値から最上部の数値を減算し（スタックは 2 段浅くなる）、その結果をスタックに積む関数である。空欄 D を埋めて関数 `subtract` を完成せよ。ただし、スタックに二つ以上の数値が積まれていない場合は、何も行わないものとする。

問 5 関数 `swap` は、図 1 (5) に示すように、スタックの最上部と上から 2 番目とを入れ替える関数である。関数 `pop` と `push` を各 2 回呼び出して数値を入れ替えることは可能であるが、`node` に割り当てたメモリの解放と獲得とを伴うため非効率である。`node` のポインタ操作により入れ替えを行う方法で、空欄 E を埋めて関数 `swap` を完成せよ。ただし、スタックに二つ以上の数値が積まれていない場合は、何も行わないものとする。

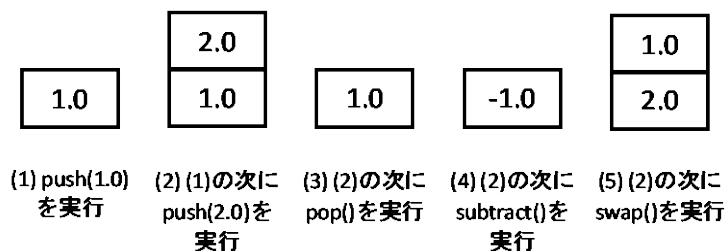


図 1 スタックの操作例

```
$ ./a.out
| 1.00 |
| 2.00 || 1.00 | | |
| 3.00 || 2.00 || 1.00 |
|-1.00 || 1.00 |
| 2.00 |
| 4.00 || 2.00 |
| 2.00 || 4.00 |
| 2.00 |
$
```

図 2 `main` 関数の実行例

問題用紙

2025	科目名	情報基礎（プログラミング）	2/3	通し番号	
------	-----	---------------	-----	------	--

プログラムリスト

```
#include <stdio.h>
#include <stdlib.h>

struct node {                /* スタックの要素 */
    double    value;         /* 数値 */
    struct node* next;       /* 次の要素 */
};

struct node* stack = NULL;   /* スタックの最上部を指す */

/* スタックの内容を最上部から下に向かって順に出力 */
void print_stack(void)
{
    struct node* nodep = stack;
    while(nodep != NULL) {
        printf("|%5.2f|", nodep->value);
        nodep = nodep->next;
    }
    printf("¥n");
}

/* スタックの深さ (要素の個数に等しい) を返す */
int stack_depth(void)
{
    struct node* nodep = stack;
    int depth = 0;
    while(nodep != NULL) {
        /* 空欄 A (この部分を作成せよ) */
    }
    return depth;
}

/* スタックに数値を積む */
void push(double top_value)
{
    struct node* nodep = (struct node*)malloc(sizeof(struct node));
    /* 空欄 B (この部分を作成せよ) */
}
```

問題用紙

2025	科目名	情報基礎（プログラミング）	3/3	通し番号	
------	-----	---------------	-----	------	--

```
/* スタックの最上部の数値を取り出す */
double pop(void)
{
    if(stack_depth() > 0){
        /* 空欄 c1 (この部分を作成せよ) */
        free(stack);
        /* 空欄 c2 (この部分を作成せよ) */
    }
    return 0.0;
}

/* スタックの 2 番目から最上部の数値を減算し、結果を積む */
void subtract(void)
{
    /* 空欄 D (この部分を作成せよ) */
}

/* スタックの最上部と上から 2 番目とを交換する */
void swap(void)
{
    /* 空欄 E (この部分を作成せよ) */
}

int main(void)
{
    /* " 1.0 2.0 3.0 - - " で 1.0 - (2.0 - 3.0) を計算 */
    push(1.0);
    print_stack();
    push(2.0);
    print_stack();
    push(3.0);
    print_stack();
    subtract();
    print_stack();
    subtract();
    print_stack();
    /* さらに" 4.0 swap - " で 4.0 - 2.0 を計算 */
    push(4.0);
    print_stack();
    swap();
    print_stack();
    subtract();
    print_stack();

    return 0;
}
```