

課題 1-1~1-4 のプログラム例

```
#include <stdio.h>

/* 配列の要素数 */
#define N 5
/* スタックに対する配列とカウンタ */
int stack[N];
int count;
/* 関数のプロトタイプ宣言 */
void clear();
void show();
void push(int x);
int pop();

/* スタックを空にする */
void clear()
{
    int i;
    /* 全ての要素に-1を代入する */
    for(i=0; i<N; i++) {
        stack[i] = -1;
    }
    /* カウンタをリセットする */
    count = 0;
}

/* スタックを出力する */
void show()
{
    int i;
    /* 各要素を出力する */
    for(i=0; i<N; i++) {
        /* dの前に空白を入れると
        正数は+の代わりに空白を出力する */
        printf("%d ", stack[i]);
    }
    printf("\n");
}

/* スタックに要素を入れる */
void push(int x)
{
    int i;
    /* スタックが一杯かどうか調べる */
    if (count<N) {
        /* 要素を1つずつ後ろにずらす */
        for(i=N-1; i>0; i--) {
            stack[i] = stack[i-1];
        }
        /* 先頭の要素にxを代入する */
        stack[0] = x;
        /* カウンタを1つ増やす */
        count++;
    } else {
        /* スタックが一杯の場合 */
        printf("Stack is full.\n");
    }
}
```

```
    }
}

/* スタックから要素を取り出す */
int pop()
{
    int i, result;
    /* 配列の先頭を戻り値とする */
    result = stack[0];
    /* スタックが空かどうか調べる */
    if (count>0) {
        /* 要素を1つずつ前にずらす */
        for(i=0; i<N-1; i++) {
            stack[i] = stack[i+1];
        }
        /* 最後の要素を-1にする */
        stack[N-1] = -1;
        /* カウンタを1つ減らす */
        count--;
    } else {
        /* スタックが空の場合 */
        printf("Stack is empty.\n");
    }
    /* 先頭の要素を戻す */
    return result;
}

int main()
{
    /* clear()の動作確認 */
    clear(); show();
    /* push()の動作確認 */
    push(5); show();
    push(8); push(3); show();
    /* pop()の動作確認 */
    printf("pop: %d\n", pop()); show();
    printf("pop: %d\n", pop()); show();
    /* カウンタを導入した場合の動作確認 */
    clear();
    push(5); push(8); push(3); push(7); push(9);
    show();
    /* スタックが一杯の場合にpushをする */
    push(4);
    show();

    printf("pop: %d\n", pop());
    printf("pop: %d\n", pop());
    printf("pop: %d\n", pop());
    printf("pop: %d\n", pop());
    show();
    /* スタックが空の場合にpopをする */
    printf("pop: %d\n", pop());
    show();

    return 0;
}
```