

게임프로그래밍(c02)

소프트웨어학과
2022764057 김도이

● 카드 - 셔플

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

#define DECK_SIZE 52

typedef struct {
    int order;
    char shape[3];
    char number[3];
} Card;

void shuffle(Card *deck) {
    int i, rnd;
    for (i = 0; i < DECK_SIZE; i++) {
        do {
            rnd = rand() % DECK_SIZE;
        } while (rnd == i); // 방법 2: 같은 위치면 다시 뽑기

        Card temp = deck[i];
        deck[i] = deck[rnd];
        deck[rnd] = temp;
    }
}
```

● 피아노 - 동요 플레이(음계)

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

void print_frequency(int octave)
{
    double do_scale = 32.7032;
    double ratio = pow(2., 1.0/12.0);
    double temp;
    int i;

    temp = do_scale * pow(2, octave - 1);
    printf("%d옥타브 : ", octave);

    for (i = 0; i < 12; i++) {
        printf("%4ld ", (unsigned long)(temp + 0.5));
        temp *= ratio;
    }
    printf("\n");
}

int calc_frequency(int octave, int inx)
{
    double do_scale = 32.7032;
    double ratio = pow(2., 1.0/12.0);
    double temp;
    int i;

    temp = do_scale * pow(2, octave - 1);
    for (i = 0; i < inx; i++) {
        temp = (int)(temp + 0.5);
        temp *= ratio;
    }
}
```

```

    }
    return (int)temp;
}

int main(void)
{
    char *scale[] = {"도","도#","레","레#","미",
                     "파","파#","솔","솔#","라","라#","시"};

    int i, octave;

    printf("음계와 주파수\n\n음계\t");
    for (i = 0; i < 12; i++)
        printf("%-5s", scale[i]);
    printf("\n");

    for (i = 0; i <= 70; i++)
        printf("-");
    printf("\n");

    for (octave = 1; octave < 7; octave++) {
        print_frequency(octave);
    }

    printf("\n[테스트] 4옥타브 미 = %d Hz\n", calc_frequency(4, 4));

    return 0;
}

```

● 자료구조 - 스택 구현(push, pop 함수 구현 및 테스트)

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

typedef struct stack_node {
    char data;
    struct stack_node *link;
} stack_node;

stack_node *top = NULL;

void push(char data)
{
    stack_node *new_node;
    new_node = (stack_node*)malloc(sizeof(stack_node));
    if (new_node == NULL) {
        printf("메모리 할당 실패\n");
        return;
    }
    new_node->data = data;
    new_node->link = top;
    top = new_node;
}

char pop(void)
{
    stack_node *temp;
    char data;
    if (top == NULL) {
        return '\0';
    }
    data = top->data;
```

```

    temp = top;
    top = top->link;
    free(temp);
    return data;
}

int main(void)
{
    int count = 0;
    char key;

    do {
        count++;
        printf("%2d 번 문자입력 >", count);
        key = getch();
        push(key);
        printf("%c \n", key);
    } while (key != 27);    // ESC 입력 시 종료

    printf("\n\n데이터 pop과정 \n\n");
    while (top != NULL) {
        key = pop();
        printf("%c ", key);
    }
    printf("\n");
    return 0;
}

```