

課題のプログラム例

```

#include <stdio.h>
#include <stdlib.h>
/* 車の情報を持つ構造体の宣言 */
typedef struct Car {
    int num;
    double gas;
    struct Car *next;
} Car;

/* 関数のプロトタイプ宣言 */
void show_carlist(Car *start, char *str);
void push_car(Car *p, int n, double g);
Car pop_car(Car *p);

/* スタックを出力する */
void show_carlist(Car *start, char *str)
{
    Car *pcar;
    printf("--- %s ---\n", str);
    for(pcar = start; pcar!=NULL;
        pcar = pcar->next) {
        printf("num: %d, gas: %lf\n",
            pcar->num, pcar->gas);
    }
}

/* スタックに要素を追加する */
void push_car(Car *p, int n, double g)
{
    Car *new;
    /* 構造体 Car の領域を確保する */
    new = (Car *)malloc(sizeof(Car));
    /* 確保した領域のメンバに引数の値を代入する */
    new->num = n;
    new->gas = g;
    /* 確保した領域の next を更新する */
    new->next = p->next;
    /* p の next は確保した領域を参照する */
    p->next = new;
}

/* スタックから要素を取り出す */
Car pop_car(Car *p)
{
    Car *first, *second;
    Car result;
    /* スタックが空の場合の処理 */
    if(p->next==NULL) {
        result.num = -1;
        result.gas = -1;
        result.next = NULL;

```

```

        return result;
    }
    /* 1 番目と 2 番目の要素を参照する */
    first = p->next;
    second = first->next;
    /* 戻り値にメンバをコピーする */
    result.num = first->num;
    result.gas = first->gas;
    result.next = NULL;
    /* 不要な要素のメモリを解放する */
    free(first);
    /* 2 番目の要素を 1 番目にする */
    p->next = second;
    /* 取り出した要素を戻す */
    return result;
}

int main(void)
{
    Car head1, car1;
    /* 空のスタックを準備する */
    head1.num = 0; head1.gas = 0;
    head1.next = NULL;
    show_carlist(&head1, "head1 (1)");
    /* push_car() の動作確認 */
    push_car(&head1, 1357, 40.3);
    push_car(&head1, 2468, 33.8);
    push_car(&head1, 3579, 26.1);
    show_carlist(&head1, "head1 (2)");
    /* pop_car() の動作確認 */
    car1 = pop_car(&head1);
    printf("\n(pop) num: %d, gas: %lf\n",
        car1.num, car1.gas);
    show_carlist(&head1, "head1 (3)");
    car1 = pop_car(&head1);
    printf("\n(pop) num: %d, gas: %lf\n",
        car1.num, car1.gas);
    show_carlist(&head1, "head1 (4)");
    /* 空のスタックから pop の動作確認 */
    car1 = pop_car(&head1);
    printf("\n(pop) num: %d, gas: %lf\n",
        car1.num, car1.gas);
    show_carlist(&head1, "head1 (5)");
    car1 = pop_car(&head1);
    printf("\n(pop) num: %d, gas: %lf\n",
        car1.num, car1.gas);
    show_carlist(&head1, "head1 (6)");

    return 0;
}

/* 課題 14-4 の main */
int main(void)
{

```

```
Car head1, car1;
int cmd, n;
double g;
/* 空のスタックを準備する */
head1.num = 0; head1.gas = 0;
head1.next = NULL;
/* 入力の説明を出力する */
printf("1: push 2: pop 3: show 4: quit\n");
cmd = 0;
while( 1 ) {
    /* 操作を入力する */
    printf("cmd > ");
    scanf("%d", &cmd);
    if(cmd == 1) {
        /* 1の場合はpush */
        printf("number > ");
        scanf("%d", &n);
        printf("gas > ");
        scanf("%lf", &g);
        push_car(&head1, n, g);
    } else if(cmd == 2) {
        /* 2の場合はpop */
        car1 = pop_car(&head1);
        printf("(pop) num: %d, gas: %lf\n",
            car1.num, car1.gas);
    } else if(cmd == 3) {
        /* 3の場合はスタックを出力 */
        show_carlist(&head1, "head");
    } else if(cmd == 4) {
        /* 4の場合は終了 */
        break;
    }
}

return 0;
}
```