

課題 2-1～2-5 のプログラム例

```
#include <stdio.h>

int main(void)
{
    int num2, num1, *max;
    int *ptr1;

    char *ptr2;
    char ch1;

    char array[5];
    int i;

    /* 【課題 2-1】 変数に初期値を入れる */
    num1 = 100;
    num2 = 200;
    /* num1 と num2 のアドレスを比較する */
    if(&num1 > &num2) {
        max = &num1;
    } else {
        max = &num2;
    }
    /* 比較した結果、大きい方の情報を出力する */
    printf("大きい方のアドレス: %p\n", max);
    printf("そのアドレスに格納されている値: %d\n", *max);

    /* 【課題 2-2】 変数とポインタに初期値を入れる */
    num1 = 1;
    ptr1 = &num1;
    /* ptr1 のみを使って 1 から 2 ずつ加算して出力する */
    for(; *ptr1<10; *ptr1+=2) {
        printf("%d ", *ptr1);
    }
    printf("\n");

    /* 【課題 2-3】 変数とポインタに初期値を入れる */
    ch1 = 'a';
    ptr2 = &ch1;
    /* ptr2 のみを使って 'a' から 'z' まで加算しながら出力する */
    for(; *ptr2<='z'; (*ptr2)++) {
        printf("%c ", *ptr2);
    }
    printf("\n");

    /* 【課題 2-4】 array[i] のアドレスを出力する */
    for(i=0; i<5; i++) {
        printf("%d: %p\n", i, &array[i]);
    }
    /* 【課題 2-5】 は array の型を変更して確認する */

    return 0;
}
```

キューのプログラム例

```

#include <stdio.h>

/* 配列の要素数 */
#define N 5
/* キューに対する配列とカウンタ */
int queue[N];
int count;
/* 関数のプロトタイプ宣言 */
void clear();
void show();
void enqueue(int x);
int dequeue();

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

/* キューを出力する */
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 ", queue[i]);
    }
    printf("\n");
}

/* キューに要素を入れる */
void enqueue(int x)
{
    int i;
    /* キューが一杯かどうか調べる */
    if (count<N) {
        /* count番目の要素にxを代入する */
        queue[count] = x;
        /* カウンタを1つ増やす */
        count++;
    } else {
        /* キューが一杯の場合 */
        printf("Queue is full.\n");
    }
}

/* キューから要素を取り出す */
int dequeue()
{
    int i, result;

    /* 配列の先頭を戻り値とする */
    result = queue[0];
    /* キュー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++) {
            queue[i] = queue[i+1];
        }
        /* 最後の要素を-1にする */
        queue[N-1] = -1;
        /* カウンタを1つ減らす */
        count--;
    } else {
        /* キューが空の場合 */
        printf("Queue is empty.\n");
    }
    /* 先頭の要素を戻す */
    return result;
}

int main()
{
    /* clear()の動作確認 */
    clear();
    show();
    /* enqueue()の動作確認 */
    enqueue(5);
    enqueue(8);
    enqueue(3);
    show();
    /* dequeue()の動作確認 */
    printf("dequeue: %d\n", dequeue());
    show();
    printf("dequeue: %d\n", dequeue());
    show();
    /* カウンタを導入した場合の動作確認 */
    /* enqueue() */
    clear();
    enqueue(5);
    enqueue(8);
    enqueue(3);
    enqueue(7);
    enqueue(9);
    show();
    /* キューが一杯の場合にenqueueをする */
    enqueue(4);
    show();
    /* dequeue() */
    printf("dequeue: %d\n", dequeue());
    printf("dequeue: %d\n", dequeue());
    printf("dequeue: %d\n", dequeue());
    printf("dequeue: %d\n", dequeue());
    printf("dequeue: %d\n", dequeue());
    show();
    /* キューが空の場合にdequeueをする */
    printf("dequeue: %d\n", dequeue());
    show();

    return 0;
}

```