

I 次のプログラム `dcount.c` は、標準入力 (キーボード) から入力された 1 行分の文字列に含まれる数字の出現頻度を表にして出力するプログラムである。実行例を参考にして、このプログラムで省略されている部分に補うべき関数 `count` の定義 (プログラム) を書きなさい。関数 `count` は第 1 引数で指定された文字列中に、第 2 引数で指定された文字コードが何度出現しているかを返り値として返す。ただし、文字コードとしては ASCII コードが用いられているものとし、入力される 1 行の文字列の内、100 バイト目以降の文字は無視されて構わない。

`dcount.c` (一部省略)

```
#include <stdio.h>

int getLine(char buf[], int size)
{
    int ch, n = 0;

    while ((ch = getchar()) != EOF && ch != '\n') {
        if (n < size - 1)
            buf[n++] = ch;
    }
    buf[n] = '\0';
    return (n == 0 && ch == EOF) ? -1 : n;
}

:

int main()
{
    char buf[100];
    int d;

    getLine(buf, sizeof(buf));
    for (d = '0'; d <= '9'; d++)
        printf("%c : %d\n", d, count(buf, d));
    return 0;
}
```

`dcount.c` のコンパイルと実行例

```
s1609h017% cc -o dcount dcount.c
s1609h017% ./dcount
12,345 times 67,890 makes 838,102,050.
0 : 4
1 : 2
2 : 2
3 : 2
4 : 1
5 : 2
6 : 1
7 : 1
8 : 3
9 : 1
s1609h017%
```

この実行例の中の「12,345 times 67,890 makes 838,102,050.」とそれに続く改行文字はキーボードから入力されたものであり、その後の 10 行はこのプログラムが出力したものである。



```

        printf(" ");
    else
        printf(" ");
    }
    printf("\n");
}

void input(struct State *st)
{
    char ch;

    while (1) {
        printf ("%s ? ", (st->turn == BLACK) ? " " : " ");
        if (scanf(" %c", &ch) != 1)
            continue;
        if (put(st, ch - 'a'))
            return;
    }
}

int main(void)
{
    struct State st;

    reset(&st);
    show(&st);
    do {
        input(&st);
        show(&st);
    } while (st.rest > 0);
    result(&st);
    return 0;
}

```

game.c のコンパイルと実行例

```

s1609h017% cc -o game game.c
s1609h017% ./game
a b c d e f g h i j

? c
a b c d e f g h i j

? d
a b c d e f g h i j

? g
a b c d e f g h i j

? f
a b c d e f g h i j

? c
a b c d e f g h i j

? d
a b c d e f g h i j

? i
a b c d e f g h i j

? h
a b c d e f g h i j

```

? a  
a b c d e f g h i j

? j  
a b c d e f g h i j

の勝ちです  
s1609h017% ./game  
a b c d e f g h i j

? d  
a b c d e f g h i j

? e  
a b c d e f g h i j

? d  
a b c d e f g h i j

? e  
a b c d e f g h i j

? g  
a b c d e f g h i j

? j  
a b c d e f g h i j

? i  
a b c d e f g h i j

? f  
a b c d e f g h i j

? b  
a b c d e f g h i j

? c  
a b c d e f g h i j

引き分けです  
s1609h017%

プログラミングおよび実習 I(中野クラス)・定期試験・解答例

```

int count(char str[], int ch)
{
    int i;
    int count = 0;

    for (i = 0; str[i] != '\0'; i++) {
        if (str[i] == ch)
            count++;
    }
    return count;
}

void reset(struct State *st)
{
    int i;

    FOR_EACH(i)
        st->line[i] = NONE;
    st->rest = LEN;
    st->turn = BLACK;
}

int put(struct State *st, int pos)
{
    int tmp;

    if (!IN(pos) || st->line[pos] != NONE)
        return 0;

    st->line[pos] = st->turn;
    st->turn *= -1;
    st->rest--;

    if (!IN(pos-1) || !IN(pos+1))
        return 1;

    if (st->line[pos+1] == st->turn && st->line[pos-1] == st->turn) {
        st->line[pos-1] *= -1;
        st->line[pos+1] *= -1;
        return 1;
    }

    tmp = st->line[pos-1];
    st->line[pos-1] = st->line[pos+1];
    st->line[pos+1] = tmp;
    return 1;
}

void result(struct State *st)
{
    int c, i;

    c = 0;
    FOR_EACH(i) {
        if (st->line[i] == BLACK)
            c++;
        else
            c--;
    }
    if (c > 0)
        printf("  の勝ちです\n");
    else if (c < 0)
        printf("  の勝ちです\n");
    else
        printf("引き分けです\n");
}

```