

小テスト

準備

プログラムを作る前に、以下の操作をしてファイルの準備をしておくこと。

1. GitHub から自分のリポジトリを clone しておく（既に今日の授業で実行済みの場合は不要）
\$ mygitclone 「自分の GitHub ユーザ名」
\$ cd prog3i-(ユーザ名)
\$./myconf
2. 今回の小テスト用のフォルダを作って移動する
\$ cd ~/prog3i-(ユーザ名)
\$ mkdir test109
\$ cd test109
3. テキストエディタでプログラムを開き、先頭行に C のコメントとして自分の番号と名前を書く
\$ gedit test109.c &

【問 1】 「入力した数値が3 の倍数以外ならば受理」するプログラムを作成して下さい。作成のポイントは以下の通りです。

- 課題 8-3 のオートマトンで、「余り 1」「余り 2」を終了状態とする
- 課題 9-1, 9-2 で作成した部分を使って、複数の終了状態で受理できるオートマトンを作る

実行結果は以下のようになります。

[実行結果]

```
$ ./a.out                                (← 10 を入力した場合の実行)
数字を入力してください。
10
読み込んだ数値 : 1 遷移先 : 1
読み込んだ数値 : 0 遷移先 : 1
受理する。
$ ./a.out                                (← 11 を入力した場合の実行)
数字を入力してください。
11
読み込んだ数値 : 1 遷移先 : 1
読み込んだ数値 : 1 遷移先 : 2
受理する。
$ ./a.out                                (← 12 を入力した場合の実行)
数字を入力してください。
12
読み込んだ数値 : 1 遷移先 : 1
読み込んだ数値 : 2 遷移先 : 0
受理しない。
```

小テストの注意点

(20 点)

- 他人の力は借りずに、自分だけでプログラムを作成する。つまり、**通常の定期試験と同様**。
- 小テスト中は、**演習室外へのネットワークアクセスは遮断される**。

小テスト中に参照できるもの

- 教科書, 配付資料
- 自分のホームディレクトリ（ホームフォルダ）以下に保存されているファイル
- * 上記以外の情報を参照することは不正行為とする
(例：USB で接続された機器に保存されているファイルの参照, ネットワークを介した情報の参照など)

答案の提出

1. 提出する全てのファイルの先頭行に、C のコメントとして自分の番号と名前を書く
2. 端末内で、以下のコマンドで課題を提出
\$ git add -A
\$ git commit -m "小テスト 9 提出"
\$ git push origin master
3. 提出が完了しているかを確認したい人は声をかけて下さい。(その場で教員側の画面で確認します)

【問 1】の模範解答

```
/* 自分の番号と名前をここに書く */
#include <stdio.h>

int isaccept(int c, int fin_states[])
{
    int i, result;
    result = 0;
    for(i=0; fin_states[i] != -1; i++) {
        if(fin_states[i]==c) return 1;
    }
    return 0;
}

int main(void)
{
    char input[100];
    int i = 0;
    int current_state = 0;
    int fin_states[3] = {1, 2, -1};

    printf("数字を入力してください。 \n");
    scanf("%s", input);

    while(input[i] != '\0'){
        switch(current_state) {
            case 0:
                if( input[i] == '0' || input[i] == '3' || input[i] == '6'
                    || input[i] == '9') { current_state = 0; }
                if( input[i] == '1' || input[i] == '4' || input[i] == '7' ) { current_state = 1; }
                if( input[i] == '2' || input[i] == '5' || input[i] == '8' ) { current_state = 2; }
                break;
            case 1:
                if( input[i] == '0' || input[i] == '3' || input[i] == '6'
                    || input[i] == '9') { current_state = 1; }
                if( input[i] == '1' || input[i] == '4' || input[i] == '7' ) { current_state = 2; }
                if( input[i] == '2' || input[i] == '5' || input[i] == '8' ) { current_state = 0; }
                break;
            case 2:
                if( input[i] == '0' || input[i] == '3' || input[i] == '6'
                    || input[i] == '9') { current_state = 2; }
                if( input[i] == '1' || input[i] == '4' || input[i] == '7' ) { current_state = 0; }
                if( input[i] == '2' || input[i] == '5' || input[i] == '8' ) { current_state = 1; }
                break;
        }
        printf("読み込んだ数値 : %c 遷移先 : %d\n", input[i], current_state);
        i++;
    }

    if(isaccept(current_state, fin_states)) {
        printf("受理する。 \n");
    } else {
        printf("受理しない。 \n");
    }
    return 0;
}
```