

実験の目的

正規表現を用いたデータ解析プログラムの設計・実装を行えるようにする。

実験の概要

正規表現を用いたデータ解析プログラムの設計・実装を行う。

課題 1

与えられたプログラム例をコンパイルし、実行した結果（スクリーンショット）を以下に添付する。

```
guest@joko01:~$ ./a.outple.txt
電話番号を入力してください：090-7623-8765
入力された電話番号は日本の携帯電話番号です。
guest@joko01:~$
```

課題 2

以下、C言語において扱うことのできる正規表現の一覧について表を記する。

メタ文字	説明
.	任意の文字
*	直前の文字を繰り返し
^	行の先頭に一致
\$	行の終端に一致
[]	囲み内の任意の文字列のうち 1 文字に一致
[^]	囲み内の任意の文字列のうちどの文字にも一致しない
+	直前の文字を 1 回以上繰り返し
?	直前の文字が 0 回か 1 回存在
{ }	直前の文字を任意の回数繰り返し
	前後のいずれかの条件に一致
()	文字を 1 つにまとめる

課題 3

課題3については報告義務がないため記載事項なしとする。

課題4

日本の電話番号を抽出し表示するプログラムを以下に記する。

```
#include <stdio.h>
#include <string.h>
#include <regex.h>

int main(int argc, char *argv[]) {
    char *regex = "^(070|080|090)-[0-9]{4}-[0-9]{4}$";
    regex_t preg;
    regmatch_t pm[2];
    char text[1026];
    FILE *fp;

    char t[24];

    if (regcomp(&preg, regex, REG_EXTENDED|REG_NEWLINE) != 0) {
        printf("regex is not available.\n");
        return 1;
    }

    fp = fopen(argv[1], "r");
    fscanf(fp, "%s", text);
    fclose(fp);

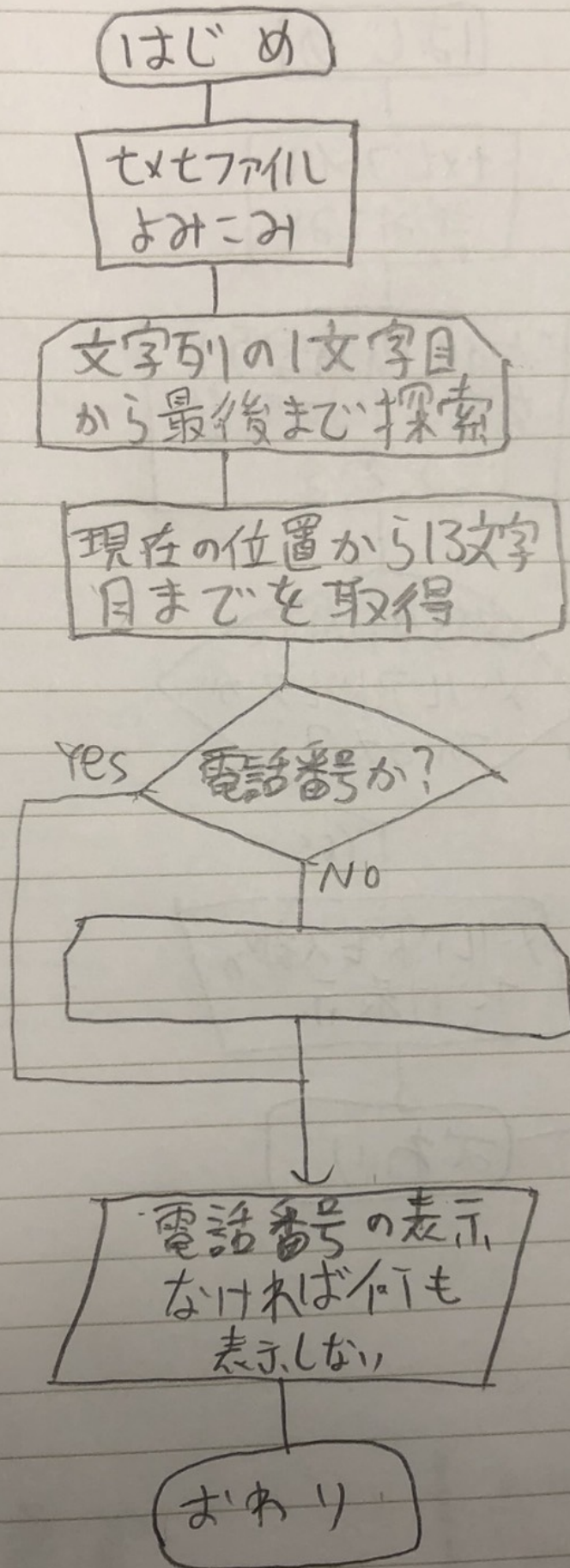
    int size = sizeof(pm) / sizeof(regmatch_t);
    int counter = 0;

    for (int i=0; i<strlen(text); i++) {
        strncpy(t, text+i, 13);
        if (regexexec(&preg, t, size, pm, 0) == 0) {
            printf("%s\n", t);
            counter++;
        }
    }

    if (counter == 0) {
        puts("マッチする電話番号はありませんでした。");
    }

    regfree(&preg);
    return 0;
}
```

以下、フローチャートを添付する。



課題 5

電子メールアドレスを抽出し表示するプログラムを以下に記する。

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

int main(int argc, char *argv[]){
    FILE *fp;
    char str[256];

    int i;

    const char *regex = "(\\w+)(\\.|_)?(\\w*)@(\\w+)(\\. (\\w+))+";
    regex_t preg;
    char str1[20];
    regmatch_t pm[1];

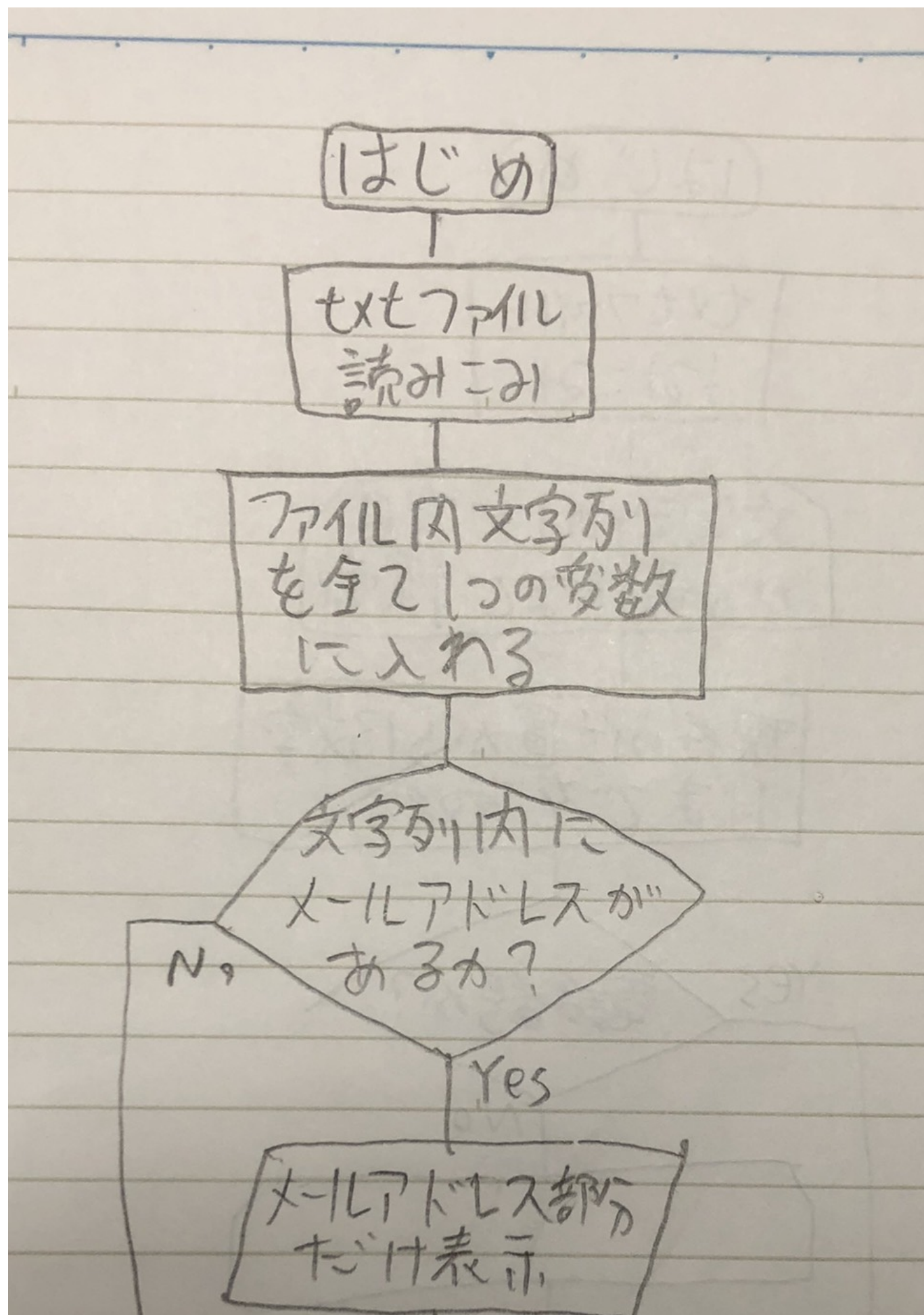
    if(regcomp(&preg, regex, REG_EXTENDED|REG_NEWLINE) != 0){
        printf("regex is not available.\n");
        return 1;
    }

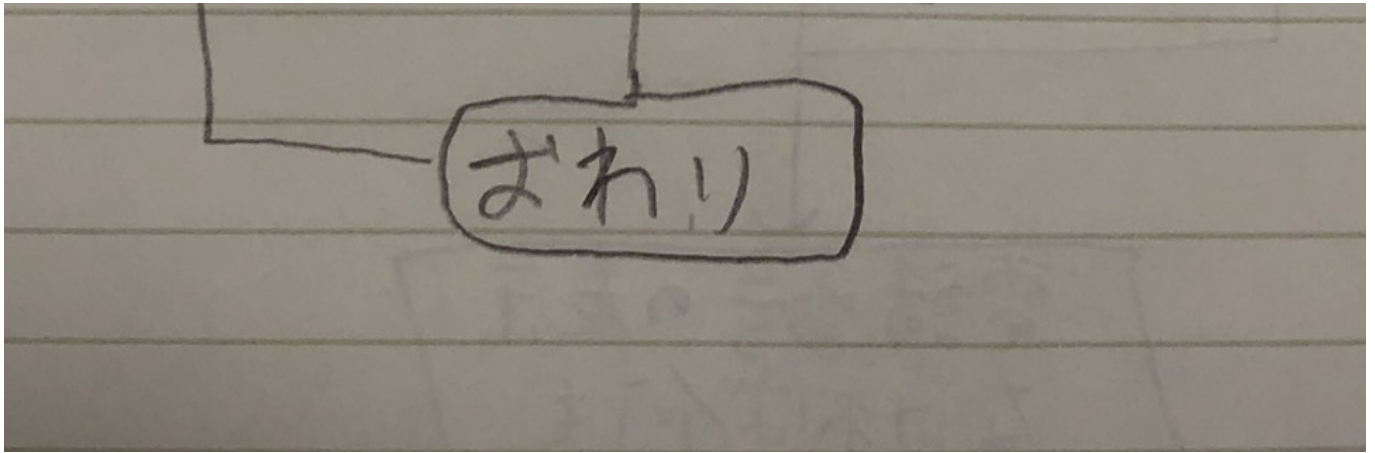
    fp = fopen(argv[1], "r");
    if(fp == NULL){
        printf("%s File Not Found\n", argv[1]);
        return -1;
    }

    while(fgets(str, 256, fp) != NULL){
        if(regexec(&preg, str, 1, pm, 0) == 0){
            i=0;
            while(str[i]!='\0') {
                if(str[i]>='a' && str[i]<='z') {
                    printf("%c",str[i]);
                } else if(str[i] == '.') {
                    printf("%c",str[i]);
                } else if(str[i] == '@') {
                    printf("%c",str[i]);
                }
                i++;
            }
            printf("\n");
        }
    }
    fclose(fp);

    regfree(&preg);
    return 0;
}
```

以下、フローチャートを添付する。





調査 1

調査事項

正規表現の歴史について調べ、どのような目的で誕生し、利用されてきたかを、200文字以上の日本語文章にまとめて報告する。

調査結果

正規表現はもともと神経科学の分野から生まれ、その後コンピュータサイエンスに応用された。

1943年に、McCullochとPittsが、人間の神経系がどのように機能するのか、あるいは機械やコンピュータが人間の脳のような働きをするように構築できるのかを説明するモデルを開発した。その後、Ken ThompsonがUNIXのテキストエディタedに正規表現を実装し、その後正規表現はコンピュータサイエンスの分野で発展していった。

今日では、正規表現はWebフォームで入力された値のバリデーションなど、幅広く使われている。

参考にしたWebページ

[Master Regular Expressions Part 1: introduction, history, engines, notation and modes](#)

[History of Regular Expression](#)

調査 2

調査事項

あなたが独自で考えた正規表現の用途について、100文字以上の日本語文章にまとめて報告しなさい。

調査結果

レポートを書く際に、書籍やWebページなどから文章を引用することがよくある。その際に正しい引用方法を用いなければ、査読者の信頼を落とすほか、そのレポート自体の品質低下につながる。

そこで、文章中の引用が正しい方法で引用されているか正規表現で判断し、もし正しくなければハイライトを付け指摘してくれるレポートエディタを提案する。

