

2021年度コンピュータ演習

第13回：1月7日

1. 授業計画
2. 練習① : 配列の使い方とソート
3. 小演習8 : 擬似乱数のソート
課題の提出 (1/12 0:00 締切り)
4. 次回の予定 (モンテカルロ法①)

本日の演習の準備（練習）

- ◆ ISTU第13回から，練習の解答例とデータをダウンロードして下さい.
- ① 配列の使い方と並べ替え
(13_pr1.cと13_data.txt)
- ◆ 保存後プログラムを実行・確認して下さい.

復習 小演習7

前回講義資料より) $-N_{\max} \sim N_{\max}$ の範囲の**浮動小数点数**を作りたい場合

$$\begin{aligned} \text{prn} &= 2.*N_{\max}*((\text{double})\text{rand}() / \text{RAND_MAX}) - N_{\max} \\ &: (-N_{\max} \leq \text{prn} \leq N_{\max}) \end{aligned}$$

小演習7: -25 以上 25 以下の**整数**の擬似乱数

誤) $\text{ri} = (\text{int})(50*((\text{double})\text{rand}()/\text{RAND_MAX}))-25;$

\Rightarrow 実行すると, $f[50] = 0$ ($\text{ri}=25$ がほとんど選択されない)

整数0は0以上1未満, 整数1は1以上2未満...というように値に幅を持っている

正) $\text{ri} = (\text{int})((2.*N_{\max}+1.)*\text{rand}() / (\text{RAND_MAX}+1.)) - N_{\max}$

$$-25 \leq r < 26$$

2. 練習①：配列の使い方とソート

- 多数の数値データを大きさの順に並べ替えたり，文字列データをアルファベット順に並べ替える操作をソーティング（Sorting）という（C-TEXT 127頁）。

【練習 1】 データファイル“13_data.txt” から，学籍番号（8桁の文字列）と数値データを，それぞれ $id[i][9]$ と $d[i]$ に読み込み，数値データ $d[i]$ が小さい順に学籍番号 $id[i][9]$ を並べ替えて画面とファイル“13_data_sorted.txt”に出力する。

- ソーティングのアルゴリズム
 - (1) 逐次比較法：最も単純な方法（C-TEXT 127頁）
 - (2) ヒープソート：(1)より能率のよい方法（129頁）

2. 練習①：配列の使い方とソート

```
// 逐次比較法
#include <stdlib.h>
#include <stdio.h>
#include <string.h>

int main() {
    int i, j, k, min, Nmax, d[150];
    char id[150][9], line[21], temp[9];
    FILE *fp;

    printf("¥n データを読み込みます. ¥n¥n");
    fp=fopen("13_data.txt", "r");
    i=0;
    while(fgets(line, 21, fp) != NULL) {
        sscanf(line, "%s %d", id[i], &d[i]);
        printf("%3d %s %5d¥n", i+1, id[i], d[i]);
        i++; // 読み込んだデータの個数を数える
    }
    fclose(fp);

    Nmax = i; // 読み込んだデータの個数
    printf("¥n データはNmax=%d個です. ¥n", Nmax);
    printf("¥n データの値が小さい順に学籍番号を並べ替えます. ¥n");
    for(i=0 ; i<Nmax ; i++){
        min = d[i];
        strcpy(temp, id[i]);
        k = i;
```

```
        for(j=i+1 ; j<Nmax ; j++){
            if(d[j] < min){
                min = d[j]; // 一時的なデータの最小値
                strcpy(temp, id[j]); // 一時的な最小値の学籍番号
                k = j; // 一時的な最小値のデータ番号(配列添字)
            }
        }
        // d[k] (min:最小値)とd[i]、id[k] (temp:最小値の学籍番号)とid[i]を入れ替える
        if(k != i){
            d[k] = d[i];
            d[i] = min; // 最小値
            strcpy(id[k], id[i]);
            strcpy(id[i], temp); // 最小値の学籍番号
        }
    }
    // 並べ替えた結果の出力(画面とファイル)
    printf("¥n 並べ替えた結果を画面に出力します. ¥n¥n");
    for(i=0 ; i<Nmax ; i++){
        printf("%3d %5d %s¥n", i+1, d[i], id[i]);
    }
    printf("¥n 並べ替えた結果をファイルに出力します. ¥n¥n");
    fp=fopen("13_data_sorted.txt", "w");
    for(i=0 ; i<Nmax ; i++){
        fprintf(fp, "%3d %5d %s¥n", i+1, d[i], id[i]);
    }
    fclose(fp);
}
```

2. 練習①：配列の使い方とソート

1. 配列の宣言

```
int d[150];      //整数型のデータ150人分：d[0]～d[149]
char line[21];   //文字列（1行分）：20文字 + '\0'（文字列の終わり）
char temp[9];    //文字列（学籍番号）：8文字 + '\0'
char id[150][9]; //文字列（150人分の学籍番号）：8文字 + '\0'
```

2. 文字列，配列データの読み込み

`fgets(line, 21, fp)` : `fp`より20文字（or 改行まで）読み込み，末尾に
'\0'を付け足して，文字の配列 `line` に格納する．

データが終了すると"NULL"を返す．

`sscanf(line, "%s %d", id[i], &d[i])` : 文字列は"%s"で読む．
変数名には"&"を付けない

3. 文字列の代入 `strcpy(a, b)` ※ `#include <string.h>` が必要．

文字列 `b[]` を `a[]` に代入（コピー）する：`strcpy(a, b)`

※ 数値の場合は，`a = b;`で代入できるが，文字列の場合は関数を用いる．

`fgets (line, 21, fp)`

◆ “13_data.txt” ファイル (fp)

B9TB4001 3137

B9TB4002 9579

B9TB4003 5643

:

B9TB4113 9803

B9TB4114 587

**`fgets (line, 21, fp)` で文字列変数の“line”に 1 行毎
文字列を読み込む。データが無くなると“NULL”を返す。**

```
sscanf(line, "%s %d", id[i], &d[i])
```

◆ “**line**” 文字列（最大21文字）から
データを変数（配列）に格納.

B9TB4001

3137

(**line**)

%s

%d



id[0][9]

d[0]

文字列

整数値

2. 練習①：ファイルの入出力

4. ファイルからデータを読み込む

- (1) `FILE *fp; // ファイル変数の宣言`
- (2) `fp = fopen("13_data.txt", "r");`
`// ファイル ("r" : 読み込み用) を開く.`
- (3) `fgets(line, 21, fp) // fpから文字列を1行読み込む.`
※ データが無くなると "NULL" を返す.
- (4) `sscanf(line, "%s %d", id[i], &d[i]); // 文字列⇒データ`
- (5) `fclose(fp); // ファイルを閉じる.`

5. ファイルへデータを書込む

- (1) `FILE *fp; // ファイル変数の宣言`
- (2) `fp = fopen("13_data_sorted.txt", "w");`
`// ファイル ("w" : 書き込み用) を開く.`
- (3) `fprintf(fp, "%3d %s¥n", d[i], id[i]);`
`// ファイルにデータを書込む.`
- (4) `fclose(fp); // ファイルを閉じる.`

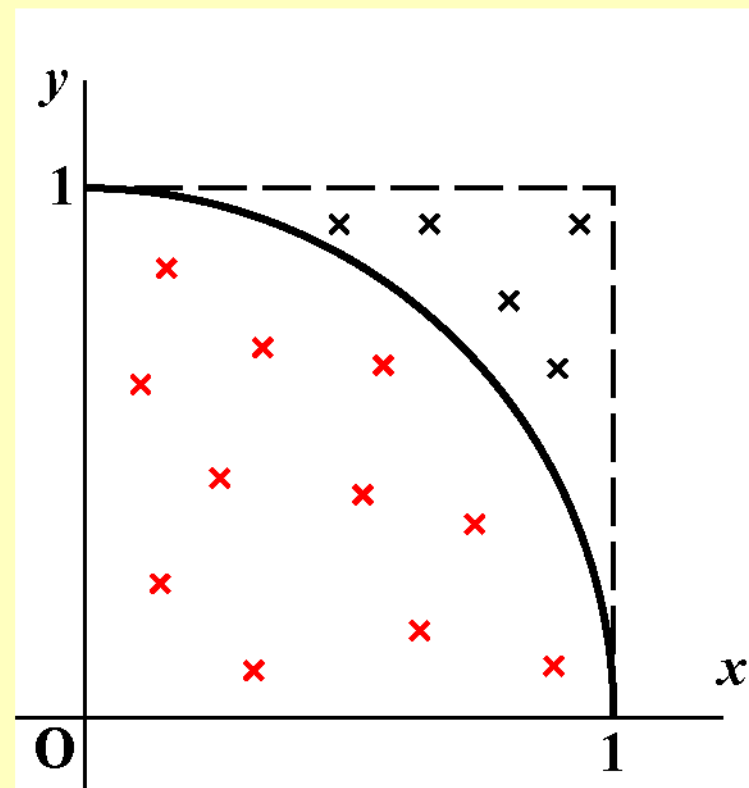
3. **小演習 8** : 擬似乱数の作成とソート

10

小演習7で作成したファイルを基に，【練習1】を参考にして， -25 から 25 の擬似乱数を**発生回数****の大きい順**に並べ替えるプログラム（学籍番号-8.c）を作成し，（順位，乱数値，発生回数）をファイル（学籍番号-8.txt）に出力せよ．

4. 次回予定：MC法による円周率の計算

- いわゆる“あたりはずれ”
(hit-or-miss) 法.
右図のような，正方形領域
($0 \leq x \leq 1, 0 \leq y \leq 1$)
内に，擬似乱数で点をとる.
得られた点が，原点を中心と
する半径 1 の1/4円内に入る
確率が， $\pi/4$ であることを利用
して，円周率 π を計算する.



図：モンテカルロ法による1/4円の数値積分

FAQ①

**Q1. 何処でエラーが発生しているのか
分かりません.**

**◆ プログラムが間違っていそうなところの
前後に,**

```
printf("\n Hit [Return] key!\n");  
getchar(); // 入力待ちのコマンド
```

**の2行を挿入してみる. (13_pr1.c参照)
⇒ 一時的に実行が中断できます.**

[Return]で再開できます.