

C言語講座

～ファイル入出力編～

ポインタ使います

例×2

練習問題×3

初めに

プログラムを組む上での話と今回の演習について

例えばこの画面の場合



背景の動画の描写

ノーツを流れてくるようにする

ボタンを押したら違反亭ライン近くのノーツは消えるようにする処理

曲の難易度ごとに判定を追加とか残りの処理

完成

演習 1 クリア

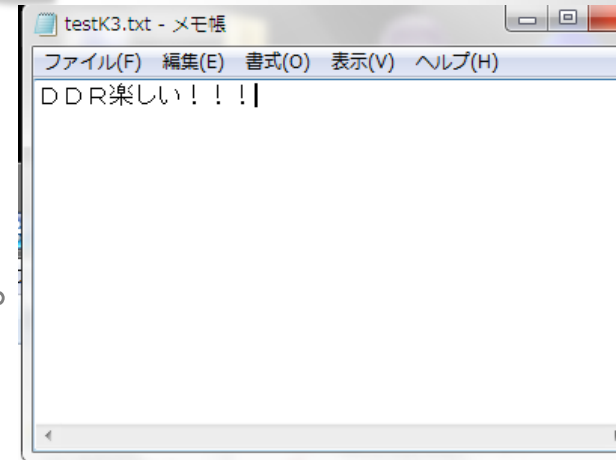
演習 2 クリア

過程は各自で考
えてください

演習 3 クリア

ファイルの出入力でこれからやること

- ファイルへの書き込み（出力）
今日はメモ帳への書き込みを行ってもらいます。
- ファイルからの読み込み（入力）
メモ帳を読みとって画面に出力してもらいます。



C言語でファイルを扱う時の ルール

1. 「ファイルのポインタを準備します！！！」

初めにファイルポインタを定義します。

2. 「ファイルを開きます！！！」

ファイルのオープンです。

関数fopenを用いて初期化です。

作業の種類によって初期化するモードが変わります（後述）

<作業>

3. 「ファイルを閉じます！！！」

ファイルのクローズです。

関数fcloseを用いてファイルポインタの解放です。

具体的に

//↓ファイルポインタの定義

```
FILE *fp;
```

//↓オープンファイルポインタ=fopen("ファイル名","モード");

```
fp = fopen("conmai.txt","w");
```

//準備完了

//↓クローズ f close(ファイルポインタ);

```
fclose(fp);
```


テキストファイルのオー プンモード

例)

```
fopen("ファイル名",モード); fopen("conmai.txt","w");
```

↑コレ

コレ↑

r : テキストファイルの読み込みモード

W : テキストファイルの書き込みモード (なかったら新規作成)

a: テキストファイルの追加モード

W+：更新用にテキストファイルをオープン（新規作成）

r+：更新用にテキストファイルをオープン（ファイルを開く）

a:更新の為、追記用にテキストファイルをオープン

関数達の簡単な紹介 1

- ・ `fprintf` : ファイルに書式付で文字列を書き込む

使い方 : `fprintf(ファイルポインタ, printfと同じ);`

例: `fprintf(fp, "%d+%d=%d", num1, num2, num3);`

- ・ `fscanf` : ファイルから文字列を読み込んで更に指定された形式に変換する

使い方 : `fscanf(ファイルポインタ, scanfと同じ);`

- 例 : `fscanf(fp, "%d", &num);`

ファイルを開いて書き込んでみよう！

テキストファイルの準備

デスクトップにある「コンピューター」あたりから
ローカルディスク C:>>users>>

ユーザー名>>desktopを開き



すると、

・・・と変わるのでこのパスをコピーしておいてください。



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

```
int main(void){
```

```
FILE*fp;
```

```
fp=fopen("C:¥¥Users¥¥ユーザー名¥¥Desktop¥¥testK3.txt
```

```
","w");
```

//さっきのファイルの場所に¥を¥に変えたもの+¥¥ファイル名

```
if(fp==NULL){
```

```
printf("ファイルオープン失敗");
```

```
return 1;
```

```
} //ファイルが開けなかったときNULLが返ってくる
```

```
fprintf(fp,"C言語講座なう ¥n"); //書き込み
```

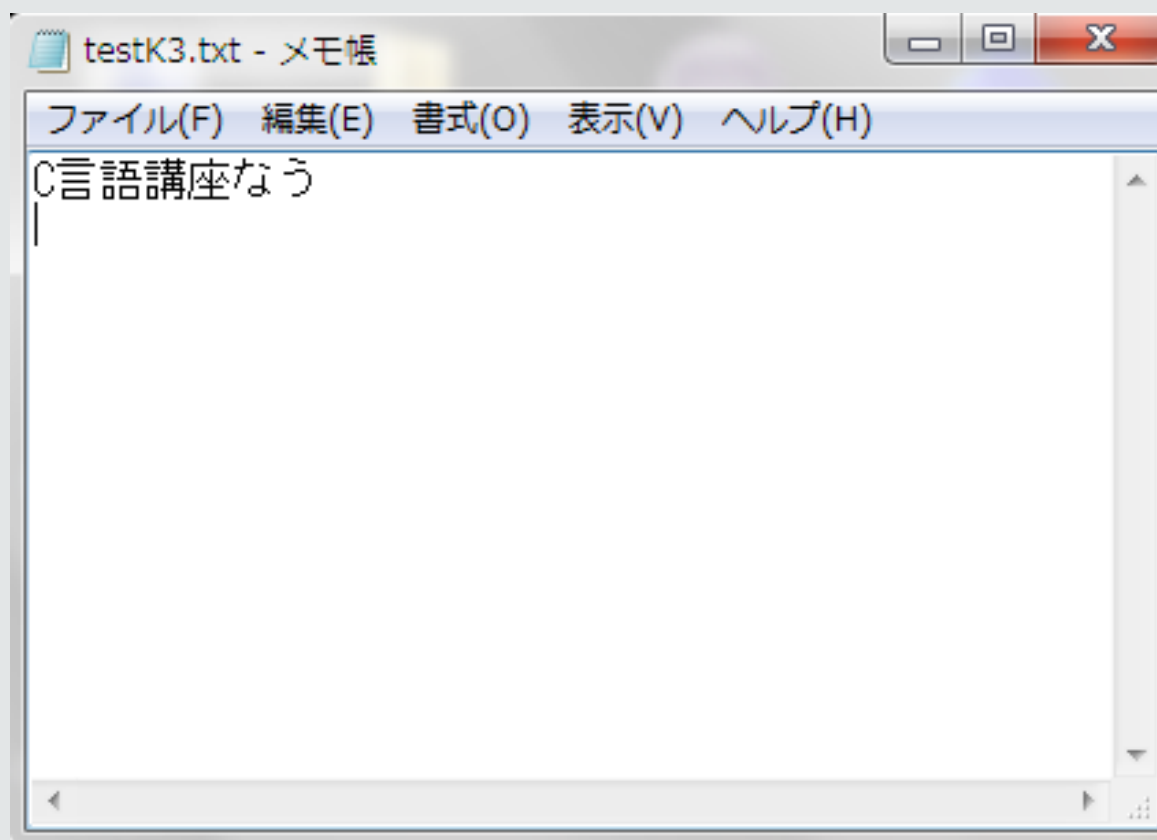
```
fclose(fp);
```

```
return 0;
```

```
}
```

実行後

デスクトップに作成されたテキストを開くと



関数達の簡単な紹介 2

- ・ `fgets` : ファイルから1行読みだす

使い方 : `fgets(アドレス, 最大文字数, ファイルポインタ);`

例 ; `char str[10];`
 `fgets( str , 10, fp );`

- ・ `fputs` : ファイルに1行書き込む

使い方 : `fputs("文字列", ファイルポインタ);`

- `fgetc`:先頭の一文字を得る。

実行後はファイルポインタが次の一文字を指す。]

使い方：`fgetc(ファイルポインタ)`:

- `fputc` : 一文字書き込む。

実行後はファイルポインタがひとつ先を指す。

使い方：`fputc(char型の文字、ファイルポインタ)`;

次は

今作成したテキストファイルを
読み込んでみましょう。

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

```
int main(void){
```

```
FILE *rfp;
```

```
char c;
```

```
rfp=fopen("(パス)~¥¥testK3.txt","r");
```

```
if(rfp==NULL){
```

```
printf("ファイルオープン失敗");
```

```
return 1;
```

```
}
```

//最後の文字まで1文字ごとに見ていく

```
while( (c= fgetc(rfp) ) != EOF ){
```

```
printf("%c",c);
```

```
}
```

```
fclose(rfp);
```

```
return 0;
```

```
}
```

(さりげなく文章を変えて)
実行すると

 C:\¥Windows¥system32¥cmd.exe

DDR楽しい！！続行するには何かキーを

演習 1

- 今作成したtestK3.txtを読み込みnewtestK3.txtという新しいテキストファイルに書き写してください。

演習のたびに次のページにヒントがあります。

ヒント

例題で出力先がコンソールだったのを
テキストファイルに変えると・・・

演習 2

Ken 183¥n
Gakut 180¥n
Miyabi 185¥n
Masaharu 181¥n
Hyde 136¥n

上の文をプログラムを通してtestK3.txtに上書きしてください。
(wモードで開けば勝手に上書きされます)

Q. testK3.txtを読み込み、それぞれの名前と身長を出力、最後にその平均をコンソールに出力してください。

ヒント

- 数字だけ読みとるにはfscanf(ファイルポインタ,"%d",&number);。一回使うことでスペース、または改行まで読み込みます。
- Char型の時fscanf()内で&は不要。

演習 3 (おそらく省略)

- testK3.txtを読み込みnewtestK3.txt
へ背の高い順に並び変えて出力し
てください。

こういう風に

Miyabi 1 8 5

Ken 1 8 3

Masaharu 1 8 1

Gakut 1 8 0

Hyde 1 3 6

ヒント

構造体の持つパラメータ例

- ・名前を保存しておくchar型の配列
- ・背の高さを保存しておくint型の変数

出力するときのヒント

```
Printf(“%s  %d”, A[n].name  ,A[n].tall );
```

プログラム組み立て例：

- ・まずはテキストを勝手に書き換えて2人分用意してただ順番を入れ替えて出力するものを作ってみる。
- ・次に背の高さだけを入れ替えて出力できるものを作ってみる。
- ・背の高さで比較して大きい方を上にして出力できるようにする。
- ・多人数verを作ってみる

答え

演習 1 答案例

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

```
//左はファイルの開くだけ
```

```
int main(void){
```

```
FILE *rfp1,*wfp2;
```

```
char c;
```

```
rfp1=fopen("(パス)¥¥testK3.txt","r");
```

```
wfp2=fopen("～パス～¥¥K3.txt","w");
```

```
if(rfp1==NULL || wfp2==NULL){
```

```
printf("ファイルオープン失敗");
```

```
return 1;
```

```
}
```

```
//rfp1を見てrfp2へ出力
```

```
while((c=fgetc(rfp1))!=EOF){
```

```
fprintf(wfp2,"%c",c);
```

```
}
```

```
fclose(rfp1);
```

```
fclose(wfp2);
```

```
return 0;
```

```
}
```

演習 2 答案例

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

```
//ファイル準備
```

```
int main(void){
```

```
FILE *rfp1;
```

```
char s[10];
```

```
int sum=0,i=0,count=0;
```

```
rfp1=fopen("～パス～¥¥testK3.txt","r");
```

```
if(rfp1==NULL){
```

```
printf("ファイルオープン失敗");
```

```
return 1;
```

```
}
```

```
while( count<5){
```

```
fscanf(rfp1,"%s",s);
```

```
fscanf(rfp1,"%d",&i);
```

```
sum =sum+i;
```

```
count++;
```

```
printf("%s",s);
```

```
printf("%d¥n",i);
```

```
}
```

```
printf("%d¥n",sum/(count+1));
```

```
fclose(rfp1);
```

```
return 0;
```

```
}
```

演習 3 答案例 1

```
include<stdio.h>
```

```
struct sourt { //構造体の宣言  
char name[20];  
int se;  
};
```

```
int main(void){  
FILE *rfp,*wfp;  
rfp=fopen("各自のパス¥¥testK3.txt","r");
```

```
int n,m,count=0;  
struct sourt rank[5],temp;
```

```
if(rfp==NULL){  
printf("ファイルオープン失敗");  
return 1;  
}  
}
```

演習 3 答案例 2

//入力

```
for(count=0;count<5;count++){  
    fscanf(rfp,"%s",rank[count].name);  
    fscanf(rfp,"%d",&rank[count].se);  
}  
fclose(rfp);
```

//並び替え

```
for(n=0;n<5;n++){//保存場  
    for(m=n;m<5;m++){//比較  
        if(rank[n].se < rank[m].se){//もし保存場よりも比較が大きければ  
            temp=rank[m];  
            rank[m]=rank[n];  
            rank[n]=temp;  
        }  
    }  
}
```

演習 3 答案例 3

```
    }  
}  
  
wfp=fopen("各自のパス¥K3.txt","w");  
if(wfp==NULL){  
    printf("ファイルオープン失敗");  
    return 1;  
}  
  
//出力  
for(n=0;n<5;n++){  
    fprintf(wfp,"%s %d¥n",rank[n].name,rank[n].se);  
}  
fclose(wfp);  
  
return 0;
```