

# prog3-1a

---

## ソースコード

```
#include <stdio.h>

int main(void){
    FILE *fp;
    char line[1000];
    fp = fopen("./prog2-1a.c", "r");
    if(fp == NULL){
        printf("ファイルがひらけませんでした\n");
        return -1;
    }
    while(fgets(line, sizeof(line), fp) != NULL){
        printf(line);
    }
    fclose(fp);
    return 0;
}
```

## 出力

```
#include <stdio.h>

#include <stdlib.h>

int main(void){

    int *array1 = (int *)malloc(sizeof(int)*10000);

    int *array2 = (int *)malloc(sizeof(int)*10000);

    int i,j,k;

    for(i=0;i<10000;i++){

        array1[i] = 2 * i - 1;

        array2[i] = 2 * i;

    }
```

```

for(j=0;j<2000;j++){

    for(i=0;i<10000;i++){

        k = array1[i];

        array1[i] = array2[i];

        array2[i] = k;

    }

}

printf("終了\n");

return 0;

}

```

## prog3-1b

---

### ソースコード

```

#include <stdio.h>

int main(void){
    FILE *fp;
    char line[1000];
    int i = 1;
    fp = fopen("./prog2-1a.c", "r");
    if(fp == NULL){
        printf("ファイルがひらけませんでした\n");
        return -1;
    }
    while(fgets(line, sizeof(line), fp) != NULL){
        printf("%d: %s", i, line);
        i++;
    }
    fclose(fp);
    return 0;
}

```

## 出力

```
1:
2: #include <stdio.h>
3: #include <stdlib.h>
4:
5: int main(void){
6:     int *array1 = (int *)malloc(sizeof(int)*10000);
7:     int *array2 = (int *)malloc(sizeof(int)*10000);
8:     int i,j,k;
9:     for(i=0;i<10000;i++){
10:         array1[i] = 2 * i - 1;
11:         array2[i] = 2 * i;
12:     }
13:     for(j=0;j<2000;j++){
14:         for(i=0;i<10000;i++){
15:             k = array1[i];
16:             array1[i] = array2[i];
17:             array2[i] = k;
18:         }
19:     }
20:     printf("終了\n");
21:     return 0;
22: }
```

## prog3-1c

---

### ソースコード

```
#include <stdio.h>

int main(void){
    FILE *fp,*fp2;
    char line[1000];
    int i = 1;
    fp = fopen("./prog2-1a.c","r");
    fp2 = fopen("./prog2-1a-line.txt","a");
    if(fp == NULL || fp2 == NULL){
        printf("ファイルがひらけませんでした\n");
        return -1;
    }
    while(fgets(line,sizeof(line),fp)!=NULL){
        fprintf(fp2,"%d: %s",i,line);
        i++;
    }
    fclose(fp);
}
```

```
fclose(fp2);
return 0;
}
```

## 出力

省略する.

## prog3-2a

---

## ソースコード

```
#include <stdio.h>

typedef struct{
    char name[100];
    int population;
} PREF;

float calcDispersion(PREF *prefs,float ave,int length){
    int i;
    float sum;
    for(i=0;i<length;i++){
        sum += (prefs[i].population - ave) * (prefs[i].population - ave);
    }
    return (float)sum / length;
}

int main(void){
    FILE *fp;
    PREF prefs[50];
    int i = 0;
    int sum = 0;
    fp = fopen("population.csv","r");
    if(fp == NULL){
        printf("ファイルが開けませんでした");
        return -1;
    }
    while(fscanf(fp,"%s %d",prefs[i].name,&prefs[i].population) != EOF){
        sum += prefs[i].population;
        i++;
    }
    printf("平均値:%f 分散:%f",(float)(sum) / (i-1),calcDispersion(prefs,(float)
(sum) / (i-1),(i-1)));
    return 0;
}
```

## 出力

平均値:2777.586914 分散:6699438.500000

## prog3-2b

---

## ソースコード

```
#include <stdio.h>

typedef struct{
    char name[100];
    int population;
} PREF;

int main(void){
    FILE *fp;
    PREF prefs[50],p;
    int i = 0,j,k;
    fp = fopen("population.csv","r");
    if(fp == NULL){
        printf("ファイルが開けませんでした");
        return -1;
    }
    while(fscanf(fp,"%s %d",prefs[i].name,&prefs[i].population) != EOF){
        i++;
    }
    for(j=0;j<i;j++){
        for(k=j+1;k<i;k++){
            if(prefs[j].population > prefs[k].population){
                p = prefs[j];
                prefs[j] = prefs[k];
                prefs[k] = p;
            }
        }
    }
    for(i=0;i<20;i++){
        printf("%s %d\n",prefs[i].name,prefs[i].population);
    }
    fclose(fp);
    return 0;
}
```

## 出力

Tottori 607  
Shimane 742  
Kochi 796  
Tokushima 810  
Fukui 822  
Saga 866  
Yamanashi 885  
Kagawa 1012  
Wakayama 1036  
Toyama 1112  
Akita 1146  
Miyazaki 1153  
Ishikawa 1174  
Oita 1210  
Yamagata 1216  
Okinawa 1362  
Shiga 1380  
Iwate 1385  
Nara 1421  
Aomori 1437

## prog3-3a

---

### ソースコード

```
#include <stdio.h>

int main(int argc, char **argv){
    FILE *fp, *fp2;
    char line[1000];
    int i = 1;
    fp = fopen(argv[1], "r");
    fp2 = fopen(argv[2], "a");
    if(fp == NULL || fp2 == NULL){
        printf("ファイルがひらけませんでした\n");
        return -1;
    }
    while(fgets(line, sizeof(line), fp) != NULL){
        fprintf(fp2, "%d: %s", i, line);
        i++;
    }
    fclose(fp);
    fclose(fp2);
    return 0;
}
```

出力

省略