目次

Q1. ［復習］以下のイ，ロ，ハは，hoge1.c，hoge2.c，hoge3.c，の「●関数定義」の部分と 1 対 1 に対応している．どれがどれに対応するか答えなさい．

イ

```
void f(double x[], int n)
{
  int i;
  for(i = 0; i < n; i++){
    x[i] = 3.1415;
  }
}
```

ロ

```
double f(double a[][10], int m, int n)
{
  double x = 0.0;
  int i, j;
  for(i = 0; i < m; i++){
    for(j = 0; j < n; j++){
      x += a[i][j];
    }
  }
  return x;
}
```

ハ

```
void f(double hoge[][10], int s, int t)
{
  int i, j;
  for(i = 0; i < s; i++){
    for(j = 0; j < t; j++){
      hoge[i][j] = 3.1415;
    }
  }
}
```

hoge1.c

```
#include <stdio.h>
● プロトタイプ宣言

int main(void)
{
  double a[10];
  int m = 5, i;

  for(i = 0; i < m; i++){
    a[i] = 2*i+1;
  }
  f(a, m);
  for(i = 0; i < m; i++){
    printf("%d %f\n", i, a[i]);
  }
  return 0;
}
```

● 関数定義

hoge2.c

```
#include <stdio.h>
● プロトタイプ宣言

int main(void)
{
  double a[10][10];
  int m = 5, n = 3, i, j;

  for(i = 0; i < m; i++){
    for(j = 0; j < n; j++){
      a[i][j] = i*j;
    }
  }
  printf("%f\n", f(a, m, n));
  return 0;
}
```

● 関数定義

hoge3.c

```
#include <stdio.h>
● プロトタイプ宣言

int main(void)
{
  double a[10][10];
  int m = 5, n = 3, i, j;

  for(i = 0; i < m; i++){
    for(j = 0; j < n; j++){
      a[i][j] = i*j;
    }
  }
  f(a, m, n);
  for(i = 0; i < m; i++){
    for(j = 0; j < n; j++){
      printf("%f\n", a[i][j]);
    }
  }
  return 0;
}
```

● 関数定義

★5　ファイル入出力

Q2. 右のソース fileio01.c をコンパイルしてできた実行ファイルが fileio01 という名前でカレントディレクトリに存在していたとする．次のようにプログラムを実行するとどのような結果が得られるか．

(1) `$ ./fileio01`

(2) `$ ./fileio01  >  piyo`

fileio01.c

```
 1  #include <stdio.h>
 2
 3  int main(void)
 4  {
 5    int x, y;
 6
 7    printf("整数を 2 つ入力してね： ");
 8    scanf("%d %d", &x, &y);
 9    printf("2 つの和は %d\n", x + y);
10    printf("ほ〜げほげほげほげらった♪\n");
11    return 0;
12  }
```

C 言語のプログラムでファイルの内容を読み書きしたい場合，以前学んだリダイレクションを使う方法が簡単である．しかし，リダイレクションでは

- プログラム中でファイル名を指定したい
- ファイル読み書きと同時に標準入出力も使いたい
- 複数のファイルを読んだり書いたりしたい

という要求に応えられない．こういう場合には，以下に述べる方法を用いる．

以下のプログラムは，fileio01.c を改造して,「2 つの和は…」という出力をファイルに対して行うようにしたものである．

fileio02.c

```
 1  #include <stdio.h>
 2  #include <stdlib.h> /* for exit() */
 3
 4  int main(void)
 5  {
 6    int x, y;
 7    FILE *fp;
 8
 9    printf("整数を 2 つ入力してね： ");
10    scanf("%d %d", &x, &y);
11
12    fp = fopen("sum", "w");
13    if(fp == NULL){
14      printf("ファイルを開けませんでした\n");
15      exit(EXIT_FAILURE); // EXIT_FAILURE は，stdlib.h で #define されている
16    }
17    fprintf(fp, "2 つの和は %d\n", x + y);
18    fclose(fp);
19
20    printf("ほ〜げほげほげほげらった♪\n");
21
22    return 0;
23  }
```

Q3. このプログラムを実行するとどのような結果が得られるか．

今度は，ファイルから入力を受け取る例を考えてみよう．以下は，fileio01.c を改造して，2 つの整数をファイルから入力させるようにしたものである．

fileio03.c

```
 1  #include <stdio.h>
 2  #include <stdlib.h> /* for exit() */
 3
 4  int main(void)
 5  {
 6    int x, y;
 7    FILE *fp;
 8
 9    fp = fopen("integer.txt", "r");
10    if(fp == NULL){
11      printf("ファイルを開けませんでした\n");
12      exit(EXIT_FAILURE);
13    }
14    fscanf(fp, "%d %d", &x, &y);
15    fclose(fp);
16
17    printf("2 つの和は %d\n", x + y);
18    printf("ほ〜げほげほげほげらった♪\n");
19
20    return 0;
21  }
```

Q4. このソースをコンパイルして，カレントディレクトリに fileio03 という名前の実行ファイルを作ったとする．また，同じディレクトリに以下の内容のファイル integer.txt があったとする．このとき，fileio03 を実行するとどのような結果が得られるか．

integer.txt

```
59 63
```

Q5. 9 行目を書き間違えて，存在しないファイル名を指定したらどうなるか．

上記の 7 行目で宣言している fp のように，ファイル入出力のためには「FILE **型** (☆ 1) **に対するポインタ**」変数を用いる．この変数は**ファイルポインタ**と呼ばれる．fopen() に成功すると，ファイルに関する様々な情報を格納した構造体変数をファイルポインタが指すようになる．ファイルを開いて読み書きができればよい，というユーザの立場としては，まずは以下のことを理解しておこう．

- ファイル入出力にはファイルポインタを使う
- fopen() はファイルがうまく開けなかったら NULL を返す

☆ 1) FILE 型というのは，stdio.h 中で定義された特別な構造体の型であり，typedef を使って型名を FILE としている．

ファイルから／への入出力の手順は，一般に次のようになる．

1. ファイルポインタ用の変数を宣言しておく　FILE [*変数名]
2. fopen() を呼んでファイルを適切なモードで開く
   [ファイルポインタ] = fopen([ファイル名], [モード指定子 (☆ 2)])
3. ファイルからの読み込み／への書き込みを行う
   printf() や scanf() などにかえて fprintf() や fscanf() などを使う
4. fclose() を呼んでファイルを閉じる　fclose([ファイルポインタ])

☆ 2) モード指定子：**"r"**は読み込みモード，**"w"**は書き込みモード．他にも，**"r+"**（更新；読み書きモード）などいろいろある

★★ 要注意 ★★ 既に存在しているファイルを "w" で fopen() すると，ファイルの内容がいったん消去され，その後の fprintf() 等の出力が新たに書き込まれる（つまり上書きされる）ことになるので注意．ファイルを読み込むつもりでうっかり fopen(”大事なファイル”, "w") とかよくやります… (;_;)　/(T_T)\