次のプログラムは外部変数に定義され、初期化された10個のint型データの平均を求め、各データと平均との差を表示するプログラムである。　このプログラムを以下のような関数の翻訳単位にファイルを分け、同様な動作をするプログラムを作成せよ。

型	関数名	引数・パラメータ	機能	戻り値
float	get_ave	なし	平均値を求め表示	平均値
void	print_data	配列の添字、平均値	1つのデータと平均値との差を求め表示	なし

＜ソースファイル＞ average_001.c

```
 1  #include <stdio.h>
 2
 3  int  score[10] = {3,5,8,9,10,6,7,9,8,3};
 4  float get_ave();
 5  void  print_data(int i,float ave);
 6
 7  int main(){
 8
 9    int sam;
10    float ave;
11    int i;
12
13    sam = 0.0;
14    sam = get_ave();
15    printf("%3.1d\n",sam);
16    ave=sam;
17    ave /= 10.0;
18    printf("average = %3.1f\n",ave);
19    for(i = 0; i < 10; i++)print_data(i, ave);
20
21
22    return(0);
23  }
24
25  float get_ave(){
26    int i;
27    float b;
28    b=0.0;
```

```
29
30    for(i=0; i<10; i++)  b+= (float)score[i];
31    return(b);
32  }
33
34  void  print_data(int i,float ave){
35    float dif = 0.0;
36    dif = score[i] - ave;
37    printf("score[%02d]=%2d  Difference from average = %4.1f\n"
38           ,i, score[i], dif);
39  }
```

＜出力結果＞

```
[nw0945:~/prog1/kadai5] e095745% ./average_001
 68
average = 6.8
score[00]= 3  Difference from average = -3.8
score[01]= 5  Difference from average = -1.8
score[02]= 8  Difference from average =  1.2
score[03]= 9  Difference from average =  2.2
score[04]=10  Difference from average =  3.2
score[05]= 6  Difference from average = -0.8
score[06]= 7  Difference from average =  0.2
score[07]= 9  Difference from average =  2.2
score[08]= 8  Difference from average =  1.2
score[09]= 3  Difference from average = -3.8
```

＜解説＞

3 行目にグローバル関数の score を定義している。score には１０個のバケツがあり数字が格納されている。４、５行目でユーザ関数 get_ave()と print_data(int i,float ave)を作り分割して計算している。関数 get_ave()では、score に入っている数字を順番に足して、メイン関数の１４行目に値を返している。合計を１７行目で割って、平均の値を計算している。ユーザ関数 print_data(int i,float ave)では平均との差をとって計算して、出力している。

＜考察＞

問題に問われていることをする前に、サンプルを検証してみたところ、makefile する足掛かりとして、前回のレポート４のようにユーザ関数を作るように分解すればいいと思った。

＜変数のスコープと記憶域クラスの考察＞

3 行目に定義した score はプログラムの全体で使用されている。これはメイン関数よりも前に定義しているためで、関数の中で作った変数をローカル変数というが、ローカル変数はその関数の中にたいしてグローバル変数は全部に適用されます。

＜ソースコード＞average_004.c

```
#include <stdio.h>

int  score[10] = {3,5,8,9,10,6,7,9,8,3};
float get_ave();
void  print_data(int i,float ave);

int main(){

  int sam;
  float ave;
  int i;

  sam = 0.0;
  sam = get_ave();
  printf("%3.1d\n",sam);
  ave=sam;
  ave /= 10.0;
  printf("average = %3.1f\n",ave);
  for(i = 0; i < 10; i++)print_data(i, ave);


  return(0);
}
```

＜ソースコード＞average_002.c

```
#include<stdio.h>

extern int score[10];

float get_ave(){
  int i;
  float b;
  b=0.0;

  for(i=0; i<10; i++)  b+= (float)score[i];
  return(b);
}
```

＜ソースコード＞average_003.c

```
#include<stdio.h>

extern int score[10];

void  print_data(int i,float ave){
  float dif = 0.0;
```

```
  dif = score[i] - ave;
  printf("score[%02d]=%2d  Difference from average = %4.1f\n"
         ,i, score[i], dif);
}
```

＜ソースコード＞makefile

```
make:average_004.o average_002.o average_003.o
        cc -o make average_004.o average_002.o average_003.o

average_004.o:average_004.c
        cc -c average_004.c

average_002.o:average_002.c
        cc -c average_002.c

average_003.o:average_003.c
        cc -c average_003.c
```

＜出力結果＞

```
nw0945:kadai5 e095745$ ./make
 68
average = 6.8
score[00]= 3  Difference from average = -3.8
score[01]= 5  Difference from average = -1.8
score[02]= 8  Difference from average =  1.2
score[03]= 9  Difference from average =  2.2
score[04]=10  Difference from average =  3.2
score[05]= 6  Difference from average = -0.8
score[06]= 7  Difference from average =  0.2
score[07]= 9  Difference from average =  2.2
score[08]= 8  Difference from average =  1.2
score[09]= 3  Difference from average = -3.8
```

＜解説＞

average_001.cで作ったプログラムを題意に沿って3つに分解し、makeコマンドで一つにする作業をしている。プログラムにあまり違いはないが、externを使ってaverage_004.cで定義したscoreをすべてのファイルに適用させている。

＜考察＞

細かいミスが多く、コンパイルでエラーが多発したが、なんとか題意通りまとめることができていると思う。

＜感想＞

今回バイトがあったのと中間テストがあったため思うようにプログラミングに時間を割くことができませんでした。友人にたくさん迷惑をかけて、なんとか作ることができました。友人に本当に感謝してもしきれません。

時間の使い方をもっとしっかりしたいと思いました。

＜参考文献＞

C 実践プログラミング