プログラミング１

Report#5(再)

提出日　：2013年８月１日
所　属　：工学部情報工学科
学籍番号：e135732J
氏　名　：前城　健太郎

次のプログラムを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

1．課題プログラム

1.1 プログラムのソースコード全体

```
1   /*
2     program   : average.c
3     comments  : 結合(linkage)を用いて、平均・差を求める。
4   */
5
6   #include <stdio.h>
7
8   int score[10]={3,5,8,9,10,6,7,9,8,3};
9
10  int main(){
11    int   i;
12    float ave = 0.0, dif;
13
14    for(i=0; i<10; i++) ave = ave + (float)score[i];
15    ave = ave / 10.0;
16    printf("ave = %3.1f\n",ave);
17
18    for(i=0; i<10; i++){
19      dif = score[i] - ave;
20      printf("score[%02d]=%2d  Difference from average = %4.1f\n",i,score[i],dif);
21    }
22
23    return(0);
24  }
```

1.2 出力結果

```
ave = 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
```

1.3 プログラムの解析

a) このプログラムはグローバル変数に定義され,8 行目で int 型変数 scoer を[0]から[9]までの 10 個を宣言し,それぞれの変数に数を格納しその数の平均と平均との差を出力するようになっている.

b) 12 行目において,変数の型を”float”としている.これは,計算結果に小数点が現れることがわかるので変数を float 型,つまり「32bit 浮動小数点数」として宣言している.

c) 14 行目から 16 行目において数の平均値を計算し表示,18 行目から 20 行目においては各値と平均値との差をそれぞれ計算し表示する.

d) ユーザー関数を使用していないため,グローバル変数"score”のスコープを指定しなくても問題なく動作している.

2.　翻訳単体にファイルを分け、同様な動作をするプログラムを作成する

2.1 Makefile の制作

2.1.1 ソースコード全体

```
#
#report@5
#課題プログラムのための Makefile
#

ave_data:report@5-1.o get_ave.o print_data.o
	gcc -o ave_data report@5-1.o get_ave.o print_data.o

report@5-1.o:report@5-1.c
	gcc -c report@5-1.c

get_ave.o:get_ave.c
	gcc -c get_ave.c

print_data.o:print_data.c
	gcc -c print_data.c
```

2.1.2 解釈

a) main 関数を含むプログラム本体(report@5-1.c),平均値を計算し表示するプログラム(get_ave.c),各値と平均値との差を計算し表示するプログラム(print_data.c)として Makefile プログラムを制作した.

b) それぞれのプログラムに変更があった場合,指定したコマンドを実行しコンパイル,リンクをしてくれる.

2.2 翻訳単位にファイルを分け、同様な動作をさせる

2.2.1 すべてのファイルのソースコード

---main 関数を含む本体ファイル(report@5-1.c)

```
#include <stdio.h>

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

int main(){

  float ave1;
  ave1 = get_ave(); /*平均値を計算し表示、ave1 に平均値を格納*/
  print_data(n,ave1); /*平均値との差を計算し表示*/

  return(0);
}
```

---サブルーチン get_ave ファイル(get_ave.c)

```
#include <stdio.h>

float get_ave(){

  extern int score[10]; /*main ファイルで宣言された変数を外部から引き継ぎ*/
  int i;
  float ave = 0.0;
  float total = 0.0;

  for(i=0;i<10;i++)
    total = total + (float)score[i];
  ave = total / 10.0;
  printf("average = %4.1f\n",ave);

  return(ave);
}
```

---サブルーチン print_data ファイル(print_data.c)

```
#include <stdio.h>

void print_data(int n,float ave1){

  extern int score[10]; /*main ファイルで宣言された変数を外部から引き継ぎ*/
  int i = 0;
  float dif;

  while(i < n){
    dif = score[i] - ave1;
    printf("score[%02d]=%2d Difference from average = %4.1f\n",i,score[i],dif);
    i++;
  }

}
```

2.2.2 出力結果

```
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
```

2.2.3 考察

a) 1 項のプログラムを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した.

b) それぞれのファイルを Makefile が存在するディレクトリと同じ場所に保存し、ターミナルにおいて make コマンドを実行し,実行可能ファイルを実行した結果が上記の出力結果である.

c) 1 つのファイルを超えて変数を扱う（この場合は”score”）ので,サブルーチン get_ave,print_data 内のそれぞれ 5 行目において記憶域クラス指定子 "extern"(外部)を使うことで呼び出している.

3. 変数の範囲(scope)の考察

3.1 方針

新たに,変数に 7 を足す関数 inc()と 1 を引く関数 dec()を含むプログラムを作成し,適当な位置に置くことで,auto（自動変数）,static（静的変数）,extern（外部変数）の記憶領域クラスによる変数の範囲の動作について考察する.

3.2 複数の関数を使う場合

3.2.1 ソースコード全体

```
1   #include <stdio.h>
2
3   void inc();
4   void dec();
5
6   int a = 743;
7   int e = 5;
8   int s = 31;
9
10  int main(){
11
12    printf("a = %02d\ne = %02d\ns = %02d\n",a,e,s);
13    puts("-----");
14    puts("\ninc------");
15    inc(); /*それぞれの変数に 7 を足します*/
16    puts("\ndec------");
17    dec(); /*それぞれの変数から 2 を引きます*/
18    puts("\ninc------");
19    inc(); /*それぞれの変数に 7 を足します*/
20    puts("-----");
21    printf("a = %02d\ne = %02d\ns = %02d\n",a,e,s);
22
23    return(0);
24  }
25
26  void inc(){
27    auto   int a;
28    extern int e;
29    static int s;
30    printf("a = %02d\n",a);
31    printf("e = %02d\n",e);
32    printf("s = %02d\n",s);
33    printf("+7\n");
34    a=a+7;
35    e=e+7;
36    s=s+7;
37    printf("a = %02d\n",a);
38    printf("e = %02d\n",e);
39    printf("s = %02d\n",s);
40
41  }
42
43  void dec(){
44    auto   int a;
45    extern int e;
46    static int s;
47    printf("a = %02d\n",a);
48    printf("e = %02d\n",e);
49    printf("s = %02d\n",s);
50    printf("-2\n");
51    a=a-2;
52    e=e-2;
53    s=s-2;
54    printf("a = %02d\n",a);
55    printf("e = %02d\n",e);
56    printf("s = %02d\n",s);
57
58  }
```

3.2.2 出力結果

```
a = 743
e = 05
s = 31
-----

inc------
a = 00
e = 05
s = 00
+7
a = 07
e = 12
s = 07

dec------
a = 00
e = 12
s = 00
-2
a = -2
e = 10
s = -2

inc------
a = 00
e = 10
s = 07
+7
a = 07
e = 17
s = 14
-----
a = 743
e = 17
s = 31
```

3.2.3 考察

a) 変数に 7 を足す関数 inc(),変数から 2 を引く関数 dec()を用いて,3 つのグローバル変数 a,e,s の中に格納されている数を操作した.

b) それぞれの関数内では,変数 a,e,s の範囲を auto,extern,static として扱い,変数を操作する前後の変数内に格納されている数を表示した.

c) auto 範囲について,変数 a はユーザー関数毎に初期値 0 が格納されており,再び同じ関数を実行しても初期値が 0 となる.

d) extern 範囲において,変数 e は異なる関数をも超えて格納されている数を扱うことができ,数も保持される.

e) static 範囲において,変数 s は異なる関数を超えて扱うことはできないが,同じ関数内でああれば一度格納された数を扱うことができる.ただし,数は保持されない.

3.3 関数を入れ子した場合使う場合

3.3.1 ソースコード全体

```
#include <stdio.h>

void inc();
void dec();

int a = 743;
int e = 5;
int s = 31;

int main(){

  printf("a = %02d\ne = %02d\ns = %02d\n",a,e,s);
  puts("-----");
  puts("\ninc------");
  inc(); /*それぞれの変数に 7 を足し 2 を引きます*/
  puts("\ndec------");
  dec(); /*それぞれの変数から 2 を引きます*/
  puts("\ninc------");
  inc(); /*それぞれの変数に 7 を足し 2 を引きます*/
  puts("-----");
  printf("a = %02d\ne = %02d\ns = %02d\n",a,e,s);

  return(0);
}

void inc(){
  auto   int a;
  extern int e;
  static int s;
  printf("a = %02d\n",a);
  printf("e = %02d\n",e);
  printf("s = %02d\n",s);
  printf("+7\n");
  a=a+7;
  e=e+7;
  s=s+7;
  printf("a = %02d\n",a);
  printf("e = %02d\n",e);
  printf("s = %02d\n",s);
  printf("-Re-dec---\n");
  dec(); /*dec()関数を inc()関数内に挿入*/

}

void dec(){
  auto   int a;
  extern int e;
  static int s;
  printf("a = %02d\n",a);
  printf("e = %02d\n",e);
  printf("s = %02d\n",s);
  printf("-2\n");
  a=a-2;
  e=e-2;
  s=s-2;
  printf("a = %02d\n",a);
  printf("e = %02d\n",e);
  printf("s = %02d\n",s);

}
```

3.3.2 出力結果

```
a = 743
e = 05
s = 31
-----

inc------
a = 00
e = 05
s = 00
+7
a = 07
e = 12
s = 07
-Re-dec---
a = 00
e = 12
s = 00
-2
a = -2
e = 10
s = -2

dec------
a = 00
e = 10
s = -2
-2
a = -2
e = 08
s = -4

inc------
a = 00
e = 08
s = 07
+7
a = 07
e = 15
s = 14
-Re-dec---
a = 00
e = 15
s = -4
-2
a = -2
e = 13
s = -6
-----
a = 743
e = 13
s = 31
```

3.3.3 考察

a) inc()関数に dec()関数を入れ子して実行した.
b) 出力結果より,入れ子をした場合,つまり関数の中に関数を入れた場合でも範囲 auto はその関数内だけでしか範囲を持たず,extern はすべての関数で変数を共有でき,static は同じ関数内であれば一度関数が終了しても格納されてる数は保存される.

3.4 図による範囲の考察

3.4.1auto について

```
main(){
int a;

inc(){
auto int a;

}

dec(){

}

inc(){

}

return(0);
}
```

a) 記憶域クラス指定子 auto の範囲は局所であり,関数が終了すると格納されている数は開放される動的記憶である.
b) また,auto は無結合であり同じ関数名であっても格納されている数を扱うことはできない.

3.4.2 extern の範囲

```
main(){
int a;

 inc(){
 extern int a;

 }

 dec(){
 extern int a;

 }

 inc(){
 extern int a;

 }

return(0);
}
```

a) 記憶域クラス指定子 extern の範囲は大域であり,関数が終了しても格納されている数は確保される静的記憶である.
b) また,extern は外部結合であり異なる関数でも格納されている数を扱うことができる.

3.4.3 static の範囲

```
main(){
int a;

 inc(){
 static int a;

 }

 dec(){
 static int a;

 }

 inc(){
 static int a;

 }

return(0);
}
```

a) 記憶域クラス指定子 static の範囲は局所または大域であり,関数が終了しても格納される静的記憶であるが,同じファイル内(関数)でしか扱うことができない.
b) static は内部結合であり同じ関数であれば格納されている数をあつかうことができる.

4.　同様な動作をするプログラムの作成

4.1 すべてのファイルのソースコード

---Makefile ファイル(Makefile)

```
#
#report@5
#課題プログラムのための Makefile
#

ave_data:report@5-1.o get_ave.o print_data.o scan.o stsc.o stde.o rank.o
	gcc -o ave_data report@5-1.o get_ave.o print_data.o scan.o stsc.o stde.o rank.o

report@5-1.o:report@5-1.c
	gcc -c report@5-1.c

get_ave.o:get_ave.c
	gcc -c get_ave.c

print_data.o:print_data.c
	gcc -c print_data.c

scan.o:scan.c
	gcc -c scan.c

stsc.o:stsc.c
	gcc -c stsc.c

stde.o:stde.c
	gcc -c stde.c

rank.o:rank.c
	gcc -c rank.c
```

---main 関数を含む本体ファイル(report@5-1.c)

```
#include <stdio.h>

int score[100];
int n; /*n=添字*/
float ave1; /*ave1 = 平均*/
float r=0.0; /*r = 標準偏差*/

void scan();
float get_ave();
void print_data(int,float);
float sts(int);

int main(){

  scan(); /*各値を変数 score に格納*/
  puts("-----");
  ave1 = get_ave(); /*平均値を計算し表示、ave1 に平均値を格納*/
  puts("-----");
  print_data(n,ave1); /*平均値との差,偏差値を計算し表示*/

  return(0);
}
```

---サブルーチン scan ファイル(scan.c)

```
#include <stdio.h>

void scan(){

  extern int score[100]; /*main ファイルで宣言された変数 score を引き継ぎ*/
  extern int n;
  int p,h;

  printf("n ==>");
  scanf("%d",&n);
  h=n;
  p = 0;
  while(0 < h){
    printf("score[%02d] ==> ",p);
    scanf("%d",&score[p]);
    p++;
    h--;
  }

}
```

---サブルーチン get_ave ファイル(get_ave.c)

```
#include <stdio.h>

float get_ave(){

  extern int score[100]; /*main ファイルで宣言された変数を外部から引き継ぎ*/
  extern int n;
  int i;
  float ave = 0.0;
  float total = 0.0;

  for(i=0;i<n;i++)
    total = total + (float)score[i];
  ave = total / n;
  printf("average = %4.1f\n",ave);

  return(ave);
}
```

---サブルーチン print_data ファイル(print_data.c)

```
#include <stdio.h>

float stde();
float stsc(int);
char rank(int);

extern int score[100]; /*main ファイルで宣言された変数を外部から引き継ぎ*/
float dif,s,r;
int i;
char c;

void print_data(int n,float ave1){

  r = stde();  /**標準偏差を計算し表示*/
  puts("-----");

  /**平均との差，偏差値を計算し表示******************/
  i = 0;
  while(i < n){
    dif = score[i] - ave1;
    if(r == 0) /*標準偏差が 0 つまりすべての score が等しい場合*/
      s = 50; /*偏差値の計算で不定形になるのを防ぐために偏差値を 50 とする*/
    else
      s = stsc(i); /*score[i]の偏差値を計算*/
    c = rank(s);
    printf("score[%02d]:Difference from average = %4.1f\n",i,dif);
    printf("the deviation = %4.1f\n",s);
    printf("Rank = %c\n",c);
    i++;
  }
  /***************************************/

}
```

---サブルーチン stsc ファイル(stsc.c)

```
float stsc(int i){

  extern float ave1,r;
  extern int score[100];
  int m;

  /**偏差値を計算*************/
  m = (((score[i]-ave1)*10)/r)+50;

  return(m);
}
```

---サブルーチン stde ファイル(stde.c)

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

extern int score[100],n;
extern float ave1;
int m;
float r;

float stde(){
  /**標準偏差を計算し表示*********/
  for(m=0;m<n;m++)
    r = r + (score[m]-ave1)*(score[m]-ave1);
  r = sqrtf(r/n); /*sqrt...平方根の計算*/
  printf("Standard Deviation = %4.1f\n",r);

  return(r);
}
```

---サブルーチン rank ファイル(rank.c)

```
1   char c;
2
3   char rank(int s){
4     /*偏差値の評価をする*/
5     if(47 <= s && s <= 53)
6       c = 'C';
7     else if(53 < s && s <= 65)
8       c = 'B';
9     else if(65 < s && s < 70)
10      c = 'A';
11    else if (70 <= s)
12      c = 'S';
13    else if(35 < s && s <= 47)
14      c = 'D';
15    else
16      c = 'E';
17
18    return(c);
19  }
```

4.2.1 出力結果　※scanf()関数には順番に 5,35,57,85,65,67 を入力した.

```
n ==>5
score[00] ==> 35
score[01] ==> 57
score[02] ==> 85
score[03] ==> 65
score[04] ==> 67
-----
average = 61.8
-----
Standard Deviation = 16.2
-----
score[00]:Difference from average = -26.8
the deviation = 33.0
Rank = E
score[01]:Difference from average = -4.8
the deviation = 47.0
Rank = C
score[02]:Difference from average = 23.2
the deviation = 64.0
Rank = B
score[03]:Difference from average =  3.2
the deviation = 51.0
Rank = C
score[04]:Difference from average =  5.2
the deviation = 53.0
Rank = C
```

4.2.2 出力結果　※scanf()関数には順番に 5,637,753,912,583,546 を入力した.

```
n ==>5
score[00] ==> 637
score[01] ==> 753
score[02] ==> 912
score[03] ==> 583
score[04] ==> 546
-----
average = 686.2
-----
Standard Deviation = 132.8
-----
score[00]:Difference from average = -49.2
the deviation = 46.0
Rank = D
score[01]:Difference from average = 66.8
the deviation = 55.0
Rank = B
score[02]:Difference from average = 225.8
the deviation = 67.0
Rank = A
score[03]:Difference from average = -103.2
the deviation = 42.0
Rank = D
score[04]:Difference from average = -140.2
the deviation = 39.0
Rank = D
```

4.3 考察

a) 今までのプログラム,考察を参考に新たに同様な動作をするプログラムを作成した.

b) 新たに追加した機能として,次のようなものがある.
(ア) 100 までの任意の数の数値を scanf 関数により入力し,計算する.
(イ) 標準偏差を計算し表示
(ウ) 標準偏差を元に偏差値を計算し表示
(エ) 偏差値を元に評価(S,A,B,C,D,E)をつける

c) プログラムを作るにあたって,新たに scan()関数,stde()関数,stsc()関数,rank()関数を定義し,main()関数を含むファイルと print_data ファイルにコードを書き足した.

d) scan()関数では,ユーザーに計算したい数値の数を入力させ(9-10 行目),その数だけ繰り返し数値を入力しその数値を変数 score[ ]へ格納していく(13-18 行目).

e) stde()関数では標準偏差を計算し表示,stsc()関数ではその標準偏差の値を extern 範囲によって用いて偏差値を計算する機能を持つ.

f) rank()関数では,偏差値の値によって評価付けを行う機能を持つ.

g) print_data では,平均値との差を計算し表示する機能に加えて,標準偏差を stde()関数を用いて計算して表示し(14 行目)stsc()関数によってそれぞれの score の値の偏差値を計算し表示している.

h) また,score に入力された数がすべて等しい場合,標準偏差が 0 となり偏差値を計算することができないので,偏差値を 50 としている(21-22 行目).