

第 10 回課題 10

(関数 II(ローカル変数とスコープ)、2010/6/24(木))

基本問題

10-1:組み合わせの数

本提出期限 2010/6/24(木)22:00

再提出期限 2010/7/8(木)14:30

提出物 : Makefile、ソースファイル (combination.c)、入力ファイル (combination.in)、出力ファイル (combination.out)

組み合わせの数を求めるプログラムを作成せよ。次の囲みに注意し、後の要求仕様を全て満たすこと。

n 個の要素を持つ集合 S の中から m 個の要素を選ぶ組合せの数 ${}_nC_m$ は、階乗関数により、次式で計算できる。

$${}_nC_m = \frac{n!}{m! \times (n - m)!}$$

要求仕様 1(全体的な要求)

- 入力は以下を満たす。(実行例参照)
 - － 全て標準入力から入力される。
 - － 集合 S の要素数を表す整数値 n (${}_nC_m$ の n) が入力される。 n は 0 以上 10 以下とする。
 - － 次に、選びだす要素数を表す整数値 m (${}_nC_m$ の m) が入力される。 m は 0 以上 10 以下でしかも $m \leq n$ とする。
- 出力は以下を満たす。(実行例参照)
 - － 全て標準出力へ行なう。
 - － 入力値が上の要求を満たす場合には、入力値 n 、 m および n 個の要素を持つ集合から m 個を選ぶ組み合わせの数 ${}_nC_m$ を整形して出力する。これら全て整数値である。
 - － 入力値が上の要求を満たさない場合には、エラーメッセージを出力し終了する。
- プログラム内部仕様として以下を満たすこと。
 - － 一つの整数値 n に対し、その階乗 $n!$ 求める関数 **factrial** を作成せよ。ただし、この関数は後の仕様を満たすこと。
 - － 二つ整数値 n, m から組合せ ${}_nC_m$ 求める関数 **combi** を作成せよ。ただし、この関数は後の仕様を満たすこと。
 - － 入力値のチェックを **main** 関数で行なうこと。
 - － 関数 **combi** を **main** 関数で呼び出し、組合せの数を標準出力に出力せよ。

要求仕様 2(関数 `factrial` への要求)

- 関数 `factrial` は以下のプロトタイプ宣言を持つこと。

```
/*  
階乗を求める関数  
  
仮引数：  
int n 階乗を求める値 (0 以上 10 以下の整数値 (int 値))  
  
戻り値：n!(n の階乗)、整数値 (int 値)  
*/  
int factorial(int n);
```

- 仮引数には整数 n を与える。呼び出された段階で、 n の値は 0 以上、10 以下として良い。
- 戻り値は、 n の階乗の値 ($n!$) とする。戻り値は整数である。
- 関数 `factrial` 内では、標準入出力 (`printf` 文、`scanf` 文) を用いてはならない。

要求仕様 3(関数 combi への要求)

- 関数 combi は以下のプロトタイプ宣言を持つこと。

```
/*  
要素数 n の集合から m 個を選ぶ組み合わせ数 nCm を求める関数  
  
仮引数：  
int n 集合 S の要素数 (0 以上 10 以下の整数値 (int 値))  
int m 選び出す要素数 (0 以上 10 以下の整数値 (int 値))  
  
戻り値：nCm(n 個の要素から m 個選ぶ組み合わせの数)、整数値 (int 値)  
*/  
int combi(int n,int m);
```

- 仮引数は求めたい組合せの数 ${}_nC_m$ の n, m を表す整数とする。呼び出された段階で、 n の値は 0 以上 10 以下として良い。 m の値は 0 以上 10 以下でしかも n 以下として良い。
- 戻り値は、 ${}_nC_m$ の整数値とする。
- 関数 combi 内では、標準入出力 (printf 文、scanf 文) を用いてはならない。

実行例

```
b11b0xx@tyy:~/prog/10$  
組合せの数 nCm を計算します。  
n(0-10) を入力して下さい。n=?  
10  
m(0-10) を入力して下さい。m=?  
3  
10C 3 =    120  
b11b0xx@tyy:~/prog/10$
```

応用問題

10-2:授業の時限

本提出期限 2010/7/1(木)14 : 30

再提出期限 2010/7/8(木)14:30

提出物 : Makefile、ソースファイル (jigen.c)、入力ファイル (jigen.in)、出力ファイル (jigen.out)

時刻から秋田県立大学の授業時間の時限を求めるプログラムを作成せよ。ただし、求める時限は、1 時限から 5 時限までとする。次の囲みに注意し、後の要求仕様を全て満たすこと。

秋田県立大学において、授業時間の時限は次のように定められている。

1 時限 8:50-10:20

2 時限 10:30-12:00

3 時限 12:50-14:20

4 時限 14:30-16:00

5 時限 16:10-17:40

要求仕様 1(全体的な要求)

- 入力は以下を満たす。
 - － 全て標準入力から入力される。
 - － 最初に、24 時間制の時を表す整数 (0-23) が入力される。
 - － 次に、分を表す整数 (0-59) が入力される。
- 出力は、以下を満たす。実行例参照。
 - － 全て標準出力へ行なう。
 - － 入力時刻が授業を行なっている時間帯であれば、入力時刻および対応する時限を出力する。
 - － 入力時刻が授業を行なっていない時間帯であれば、入力時刻および適切なメッセージを出力する。
 - － 入力された 2 つの整数が時刻の形式でなければ、適切なメッセージを出力する。例えば、時間として負の値あるいは 24 以上の値が入力された場合や、分として負の値あるいは 60 以上の値が入力された場合には、適切なメッセージを出力する。
- プログラム内部仕様として以下を満たすこと。
 - － 入力された 2 つの整数に対して、`main` 関数内で時刻の形式に沿っているかどうかの判定を行なうこと。
 - － 時刻を表わす 2 つの整数から、時限を求める関数 `time_to_jigen` を作成せよ。ただし、この関数は後の仕様を満たすこと。
 - － 関数 `time_to_jigen` を `main` 関数で呼び出し、入力時刻が授業時間中かを判定し、授業時間中であれば対応する時限を求める。

要求仕様 2(関数 `time_to_jigen` への要求)

- 関数 `volume_ball_radius` は以下のプロトタイプ宣言を持つこと。

```
/*  
時刻から時限を求める関数  
  
仮引数：  
hour:時間 (24 時間制、0-23 の整数)  
minute:分 (0-59 の整数)  
  
戻り値：  
授業時間帯以外や時刻の形式に合わない場合:0  
授業時間帯:対応する時限 (1-5)  
*/  
int time_to_jigen(int hour,int minute);
```

- 仮引数 `hour` と `minute` は、24 時間制の時刻が与えられるとする。
- 戻り値として、仮引数 `hour` と `minute` で表わされた時刻が、授業時間帯以外や時刻の形式にあわなければ 0 を、授業時間中であれば対応する時限を返す。
- 関数 `time_to_jigen` の関数定義部分では、標準入出力等を行ってはならない。

実行例 1

```
b11b0xx@tyy:~/prog/10$./jigen
秋田県立大学において、時刻から時限を求めます。
時間 (0 - 23)?
8
分 (0 - 59)?
55
時刻 8 : 55 は、第 1 時限です。
b11b0xx@tyy:~/prog/10$
```

実行例 2

```
b11b0xx@tyy:~/prog/10$./jigen
秋田県立大学において、時刻から時限を求めます。
時間 (0 - 23)?
10
分 (0 - 59)?
25
時刻 10 : 25 は授業時間外です。
b11b0xx@tyy:~/prog/10$
```

実行例 3

```
b11b0xx@tyy:~/prog/10$./jigen
秋田県立大学において、時刻から時限を求めます。
時間 (0 - 23)?
25
時間は、(0 - 23) の範囲にして下さい。
b11b0xx@tyy:~/prog/10$
```