

第一版 第一刷

C言語問題集

上級問題

12月30日更新

目次

• 情報	2
• 問題 1	3
• 問題 2	5
• おまけの問題用紙	11
• 解答例	19

上級問題

情報

内 容 : C言語

問題形式 : (全問) プログラム記述式問題

解 答 : 問に正対するプログラムを記述し、正しい動作結果を得られたら正解になります。

難 易 度 : 上級 (標準レベルの内容とポインタ、構造体、ファイル入出力ができる程度)

問 題 数 : 2問

最終更新 : 2014年12月30日

解 答 例 : 2015年1月20日公開予定

目標時間 : 4時間

(サイトより引用)

1

入力された金額を金種別に分類するプログラムを作成しなさい。但し、次の条件に合うように金種計算プログラムを完成させよ。

条件：

- ①枚数は、考えられる数の最小になるようにすること。また今回は、一万円、五千元、千円、五百円、百円、五十円、十円、五円、一円を考慮すること。（二千元は考えない）
- ②プログラムは、最大10人分のデータが入るようにすること。
- ③プログラムでは構造体を用いてデータを作成すること、データの内訳は、図1-1のようにすること。
- ④一連の処理が完了したとき、「別データを入力しますか？」と問い、ユーザーがYES(1)を入力したら、入力処理から再開し、NO(0)が入力されたら終了するようにしなさい。その処理は、図1-2の処理文を利用し、main関数内に記述すること。（文中の変数名は変更しても構わない）
- ⑤プログラムの自作関数は、図1-3のように構成し、main関数内に記述するのは、変数宣言と自作関数、と図1-2の処理文とそれに関連する文のみとし、それ以外の入力、演算、出力は、それぞれの自作関数内で行うこと。
- ⑥プログラム上では次の文の記述を禁止する。
 - ・goto構文とラベル文
 - ・switch構文（else ifによるスイッチも含む【if、elseは可】）
- ⑦プログラム上では次の要素を入れること。
 - ・ポインタ（[*][&]間接参照演算子等）
 - ・構造体（typedef struct ～等）
 - ・文字列配列
- ⑧出力は、画面に表示するのとtxt出力もすること。またそれぞれの出力には、それぞれ別の関数を作成すること。（呼び出しの順序は問わない）
- ⑨出力の際には、図1-4のように「一万円は、XXXX枚」のように表記し、体裁をそろえること。
- ⑩テキスト出力の際、テキストファイルには、日付、時刻データも記入すること。またそのために「time.h」をインクルードし、「tm構造体(time.hで定義されている構造体)」を使用し、「time関数、localtime関数

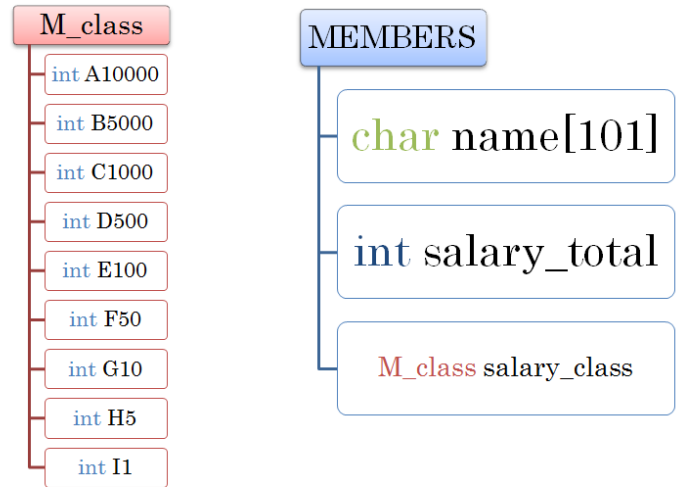
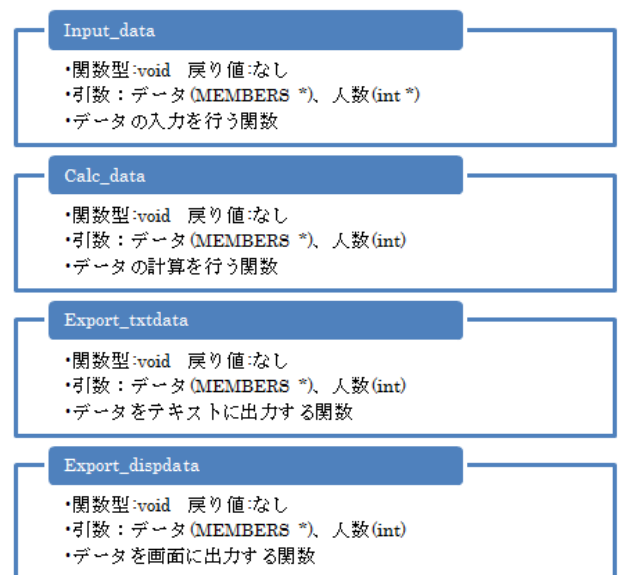


図1-1 構造体データ内訳

```
int F;
do{
    //一連の処理
    do{
        printf("別データを入力しますか? %nYes(1),NO(0)");
        scanf("%d",&F);
    }while(F!=1&&F!=0);
}while(F==1);
return 0;
```

図1-2 main関数内、終了処理の例



以上の4つの自作関数を作成する。関数型、戻り値、引数は指定通り但し () 内は変数型を示し、変数名は省略。*はポインタ（間接参照演算子）を示す。
変数名、関数名は、任意の値に変更しても構わない。

図1-3 作成する自作関数の構成、主な内容

(time.hで定義されている関数)」を使用すること。

また「time.h」を用いたプログラムの例を図1-5を参考に添付する。

⑪その他の各宣言（配列）の例として、図1-6のようにする。

⑫プログラムの例として、図1-7（画面表示）、図1-8（テキスト出力）に示す。

⑬プログラムは200行以内で記述すること。

```
int main(){
    time_t timeValue;
    struct tm *timeObject;

    time(&timeValue);
    timeObject = localtime(&timeValue);
    printf("ただいまの日付、時刻は、%d年%d月%d日 %d時%d分%d秒です。",
        timeObject->tm_year + 1900, timeObject->tm_mon + 1, timeObject->tm_mday,
        timeObject->tm_hour, timeObject->tm_min, timeObject->tm_sec);
    return 0;
}
```

注) time.hをインクルードすること

図1-5 time.hを用いた現在時刻取得の例

一万円は、	12枚
五千円は、	0枚
千円は、	3枚
五百円は、	0枚
百円は、	4枚
五十円は、	1枚
十円は、	0枚
五円は、	1枚
一円は、	1枚

図1-4 出力例

```
const int
MONEY[]={10000,5000,1000,500,100,50,10,5,1};
```

【演算するための金種の宣言】

```
static char
```

```
*Money_Ord[]={"一万円","五千円","千円","五百円","百円","五十円","十円","五円","一円"};
```

【出力時の文字列】

図1-6 関数外で宣言する各宣言（配列）の例

給料金種内訳明細書【2014年12月26日 3時27分2秒生成】.txt
給料金種内訳明細書↓

```

1人目：羽鶴太郎さんの給料205758円の金種内訳↓
一万円は、    20枚↓
五千円は、    1枚↓
千円は、      0枚↓
五百円は、    1枚↓
百円は、      2枚↓
五十円は、    1枚↓
十円は、      0枚↓
五円は、      1枚↓
一円は、      3枚↓
2人目：希条裕樹さんの給料211835円の金種内訳↓
一万円は、    21枚↓
五千円は、    0枚↓
千円は、      1枚↓
五百円は、    1枚↓
百円は、      3枚↓
五十円は、    0枚↓
十円は、      3枚↓
五円は、      1枚↓
一円は、      0枚↓

```

-----↓
-金種別累計枚数-↓

```

一万円は、    41枚↓
五千円は、    1枚↓
千円は、      1枚↓
五百円は、    2枚↓
百円は、      5枚↓
五十円は、    1枚↓
十円は、      3枚↓
五円は、      2枚↓
一円は、      3枚↓

```

-----↓
[EOF]

ファイル名：給料金種内訳明細書【2014年12月26日 3時27分2秒生成】.txt

図1-8 テキストファイル出力例

```

給料金種別計算機
入力する人数を指定してください(最大10人):2
1人目名前入力【半角100文字(全角50文字)以内、空白入力禁止】:羽鶴太郎
羽鶴太郎さんの給料入力【0以上100,000,000以下】:205758
2人目名前入力【半角100文字(全角50文字)以内、空白入力禁止】:希条裕樹
希条裕樹さんの給料入力【0以上100,000,000以下】:211835
給料金種内訳明細書

```

```

-----↓
1人目：羽鶴太郎さんの給料205758円の金種内訳
一万円は、    20枚
五千円は、    1枚
千円は、      0枚
五百円は、    1枚
百円は、      2枚
五十円は、    1枚
十円は、      0枚
五円は、      1枚
一円は、      3枚
2人目：希条裕樹さんの給料211835円の金種内訳
一万円は、    21枚
五千円は、    0枚
千円は、      1枚
五百円は、    1枚
百円は、      3枚
五十円は、    0枚
十円は、      3枚
五円は、      1枚
一円は、      0枚

```

-----↓
-金種別累計枚数-

```

一万円は、    41枚
五千円は、    1枚
千円は、      1枚
五百円は、    2枚
百円は、      5枚
五十円は、    1枚
十円は、      3枚
五円は、      2枚
一円は、      3枚

```

-----↓
別データを入力しますか?
Yes(1),NO(0) 1

この例は、2人のデータの入力を行い、その出力結果（画面）を表示した例である。

金種別累計枚数は、2人の各金種の合計である。

別データを入力しますか?の質問に対し、0ならそのまま終了、1なら処理を最初から行うようにする。（この画面では、その入力前の状態を表しています、このとき既に入力したデータは、テキストデータに出力されている）

図1-7 画面出力例

2

こいこい役判定、文数計算プログラムを作成せよ。但し、次のルール、条件に合うようにプログラムを完成させよ。

こいこい役判定、文数についてのルール

※図2-1にも絵付で説明しています。

①判定する役と文数は、次の13種類とする。

(1)、五光 (10文)

・光札5枚

(2)、四光 (8文)

・光札4枚但し「小野道風に蛙」を含まない

(3)、雨四光 (7文)

・光札4枚但し「小野道風に蛙」を含む

(4)、三光 (5文)

・光札3枚但し「小野道風に蛙」を含まない

(5)、花見で一杯 (5文)

・「菊に盃」「桜に幕」

(6)、月見で一杯 (5文)

・「菊に盃」「芒に月」

(7)、猪鹿蝶 (5文+猪鹿蝶以外のタネ札の数)

・「萩に猪」「紅葉に鹿」「牡丹に蝶」

(8)、赤短青短 (10文+赤短、青短以外の短冊札の数)

・赤短札3枚と青短札3枚

(9)、赤短 (5文+赤短以外の短冊札の数)

・赤短札3枚

(10)、青短 (5文+青短以外の短冊札の数)

・青短札3枚

(11)、タネ (1文)

・タネ札(10点札) 5枚以上、6枚目から1枚毎に+1文

(12)、タン (1文)

・短冊札5枚以上、6枚目から1枚毎に+1文

(13)、カス (1文)

・カス札10枚以上、11枚目から1枚毎に+1文

②①以外の役、手四、くつつきの判定は行わない。

③役ルールとして次の条件を考慮すること。

・(7)猪鹿蝶が成立した時点で(11)タネ消滅

・(8)赤短青短が成立した時点で(9)赤短、(10)青短、(12)タン消滅

・(9)赤短が成立した時点で(12)タン消滅

・(10)青短が成立した時点で(12)タン消滅

④札のルールについて次のように定義する。

・光札：「鶴に松」「桜に幕」「芒に月」「小野道風に蛙」「桐に鳳凰」

・タネ札(10点札)：「梅に鶯」「藤にほととぎす」「菖蒲に八ツ橋」「牡丹に蝶」「萩に猪」「芒に

雁」「菊に杯」「紅葉に鹿」「柳に燕」

- ・短冊札：芒（8月）、桐（12月）以外に1枚ずつ計10枚
- ・カス札：上記以外
- ・「小野道風に蛙」は、カス札、タネ札に計上される。

※詳細は、図2-1参照

⑤各月毎の花鳥風月を次のように定義する。

- 1月：松
- 2月：梅
- 3月：桜
- 4月：藤
- 5月：菖蒲（アヤメ）
- 6月：牡丹（ボタン）
- 7月：萩
- 8月：芒（ススキ）
- 9月：菊
- 10月：紅葉（モミジ）
- 11月：柳
- 12月：桐

⑥合計7文以上の場合は、文数は2倍になる。

⑦こいこいルールは次のURLを参考に定義するものとする。

http://www.nintendo.co.jp/n09/hana-kabu_games/

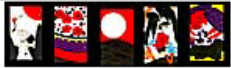





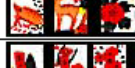


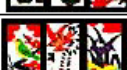

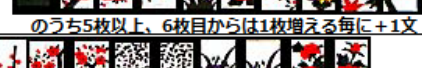

札について	役について
1月（松） <small>松に鶴</small>	五光 10文 <small>光札5枚</small> 
2月（梅） <small>梅に鶯</small>	四光 8文 <small>小野道風に蛙（柳に雨）以外の光札4枚</small> 
3月（桜） <small>桜に鶯</small>	雨四光 7文 <small>小野道風に蛙（柳に雨）含む光札4枚</small> 
4月（藤） <small>藤に不如帰</small>	三光 5文 <small>小野道風に蛙（柳に雨）以外の光札3枚</small>  <small>のうち3枚</small>
5月（菖蒲） <small>菖蒲に八つ橋</small>	花見で一杯 5文 
6月（牡丹） <small>牡丹に蝶</small>	月見で一杯 5文 
7月（萩） <small>萩に猪</small>	猪鹿蝶 5文 <small>猪鹿蝶以外のタネ札1枚につき+1文</small> 
8月（芒） <small>芒に月、芒に雁</small>	赤短青短 10文 <small>赤短青短以外の短冊札1枚につき+1文</small> 
9月（菊） <small>菊に盃</small>	赤短 5文 <small>赤短以外の短冊札1枚につき+1文</small> 
10月（紅葉） <small>紅葉に鹿</small>	青短 5文 <small>青短以外の短冊札1枚につき+1文</small> 
11月（柳） <small>小野道風に蛙、柳に燕</small>	タネ 1文 <small>小野道風に蛙もタネ札に計上する</small> <small>タネ札5枚以上で成立</small> <small>6枚以上は1枚につき+1文</small> <small>猪鹿蝶が成立したら谷消滅</small>  <small>のうち5枚以上、6枚目からは1枚増える毎に+1文</small>
12月（桐） <small>桐に風扇</small>	タン 1文 <small>短冊札5枚以上で成立</small> <small>6枚以上は1枚につき+1文</small> <small>赤短青短、赤短、青短が成立したら谷消滅</small>  <small>のうち5枚以上、6枚目からは1枚増える毎に+1文</small>
<small>赤字…光札</small> <small>緑字…タネ札</small>	カス 1文 <small>カス札10枚以上で成立</small> <small>11枚以上は、1枚につき+1文</small> <small>小野道風に蛙もカス札に計上する</small>  <small>のうち10枚以上、11枚目からは1枚増える毎に+1文</small>

図2-1こいこいの札、役について

条件：

①役と文数、札については、全て上記ルールに従うものとする。

②プログラムでは配列

(文字列配列含む)、列挙体を用いてデータを作成すること、但し列挙体は、プログラムを明確にするために使用する、データの内訳は、図2-2のようにすること。

③プログラムの処理構成は、入力、計算、出力共すべて自作関数で行うものとする。

④プログラム上では次の文の記述を禁止する。

- ・ goto 構文とラベル文
- ・ switch 構文 (else if, if, else は可)

⑤プログラム上では次の要素を入れること。

- ・ ポインター ([*][&]間接参照演算子等)
- ・ 列挙体 (enum)
- ・ 文字列配列

⑥main関数に記述するのは、タイトル「こいこい役計算機」とデータの宣言と自作関数のみとする。

⑦入力処理の関数 (Input_data) は、各月毎に取得したパターン (16通り) を明示し、数字0~15で解答させる。12回行い、データを16進数で保存するようにする。但、全てのパターンは、

281,474,976,710,656通り (=6byte) でint型では最大2,147,483,647bit (=4byte-符号ビット (1bit)) の値しか保存できないので、パターンは、(unsigned) long long型 (=8byte) を使うこと。メモリの仕様を図2-3で示す。(解答例ではunsigned long long型で記述しています。)

また、0~15以外の数字が入力されたら、再入力させること。

⑧入力処理が終了したら、取得したパターンを基に、取得した札を明示したうえで、このパターンで処理を実行するか尋ねる事。

⑨計算処理を行う関数 (Calc_Fuda) では、札種別の枚数 [13]、48枚の所持の有無 [48]、役別文数 [13] (全て配列) の値を計算すること。

⑩出力処理 (Output_data) では、計算処理で計算した、役別文数を使い、出力を行う、またその際所

```
static char *KTFG[]={{"松","梅","桜","藤","宮蒲","牡丹","萩","芒","菊","紅葉","柳","桐"}; //花鳥風月 (月順)
static char *FUDA[]={ //札の文字
    "カス","カス","赤短冊","松に鶴 (光札)", //1月 (松)
    "カス","カス","赤短冊","梅に鶯 (タネ札)", //2月 (梅)
    "カス","カス","赤短冊","桜に鶯 (光札)", //3月 (桜)
    "カス","カス","短冊","藤にホトトギス (タネ札)", //4月 (藤)
    "カス","カス","短冊","宮蒲に八つ橋 (タネ札)", //5月 (宮蒲)
    "カス","カス","青短冊","牡丹に蝶 (タネ札)", //6月 (牡丹)
    "カス","カス","短冊","萩に猪 (タネ札)", //7月 (萩)
    "カス","カス","芒に雁 (タネ札)", "芒に月 (光札)", //8月 (芒)
    "カス","カス","青短冊","菊に盃 (タネ札)", //9月 (菊)
    "カス","カス","青短冊","紅葉に鹿 (タネ札)", //10月 (紅葉)
    "カス","短冊","柳に燕 (タネ札)", "小野道風に蛙 (光札)", //11月 (柳)
    "カス","カス","カス","桐に鳳凰 (光札)", //12月 (桐)
};
static char *N_YAKU[]={ //役の名前と探査、表示順序
    "五光","四光","雨四光","三光","花見で一杯","月見で一杯","猪鹿蝶","赤短青短","赤短","青短","タネ","タン","カス"
};
static int YAKU_N[]={ //役の文数
    10,8,7,5,5,5,5,10,5,5,1,1,1
};
typedef enum { //役の名前と探査、表示順序
    五光,四光,雨四光,三光,花見で一杯,月見で一杯,猪鹿蝶,赤短青短,赤短,青短,タネ,タン,カス
}NAME_YAKU;
typedef enum { //札の名前
    カス札=1,タン札,赤短札,青短札,タネ札,猪鹿蝶札,盃札,松に鶴,桜に鶯,芒に月,小野道風に蛙,桐に鳳凰
}NAME_FUDA;
static int FUDA_NUM[]={ //札の種別
    1,1,1,3,9, //1月
    1,1,1,3,5, //2月
    1,1,1,3,9, //3月
    1,1,1,2,5, //4月
    1,1,1,2,5, //5月
    1,1,1,4,6, //6月
    1,1,1,2,6, //7月
    1,1,1,5,10, //8月
    1,1,1,4,7, //9月
    1,1,1,4,6, //10月
    1,2,5,11, //11月
    1,1,1,12 //12月
};
/*
カス札は、全て:1 count
短冊札は、無地:2、赤短:3、青短:4 count
タネ札は、梅に鶯:5、藤に不如帰:5、宮蒲に八つ橋:5、牡丹に蝶:6、萩に猪:6、芒に雁:5、菊に盃:7、紅葉に鹿:6、柳に燕:5、
光札は、松に鶴:8、桜に鶯:9、芒に月:10、小野道風に蛙:11、柳に鳳凰:12
*/
```

図2-2データの構造 (宣言例)

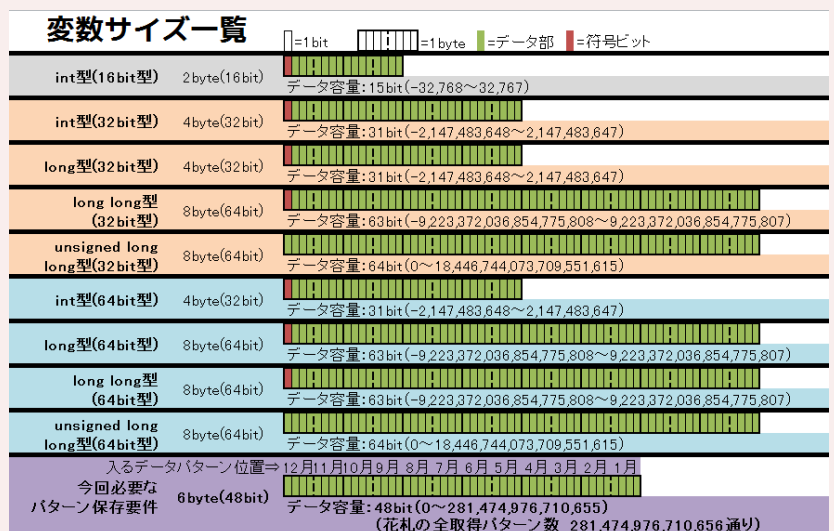


図2-3 メモリの仕様

持っている札も明示すること。

⑪所持している札を明示する際、月ごとに表示し、その時その月の花鳥風月、札の名前を明示すること。明示は、入力の最後の確認、出来役を表示するときに行う。

明示の例を図2-4で示す。

⑫関数の定義を図2-5で示す。

⑬データを格納する配列の例を図2-2で示す。

⑭図2-2データの定義、図2-5関数の定義で使った名前（関数名、配列名、列挙体名及び列挙体で定義される文字）は、変更しても構わない。

⑮図では、列挙体の文字は明確にするために日本語で表記していますが、コンパイラによっては日本語表記ができない場合があるので、その場合は、ローマ字や英語表記で記述しても構わない。

⑯出力結果の例を図2-7で示す

⑰プログラムは、300行以内で記述すること。

```
1月 (松) の札で所持してる札: 「カス」 「カス」 「赤短冊」
2月 (梅) の札で所持してる札: 「カス」 「カス」
3月 (桜) の札で所持してる札: 「カス」 「赤短冊」
4月 (藤) の札で所持してる札: 「カス」 「短冊」
5月 (菖蒲) の札で所持してる札: 「カス」 「カス」 「短冊」
6月 (牡丹) の札で所持してる札: 「カス」 「牡丹に蝶 (タネ札)」
7月 (萩) の札で所持してる札: 「短冊」 「萩に猪 (タネ札)」
8月 (芒) の札で所持してる札: 「カス」 「芒に月 (光札)」
9月 (菊) の札で所持してる札: 「カス」
10月 (紅葉) の札で所持してる札: 「カス」
11月 (柳) の札で所持してる札: 「カス」 「短冊」 「柳に燕 (タネ札)」 「小野道風に蛙 (光札)」
12月 (桐) の札で所持してる札: 「桐に鳳凰 (光札)」
```

※図2-6「実行例（簡略版）」は、ここでは省略する

図2-4 所持札表示の例



以上の4つの自作関数を作成する。関数型、戻り値、引数は指定通り
但し () 内は変数型を示し、変数名は省略。*はポインタ（間接参照演算子）を示す。

変数名、関数名は、任意の値に変更しても構わない。

図2-5 作成する自作関数の構成、主な内容

こいこい設計算機
 違った札を月ごとに0～15の数値で入力してください。
 1月の札(松)で所持してる札の候補を次から選んで数値0～15で入力してください。
 (同一候補がある場合、候補内でどの値を入れても構いません)
 0:所持なし
 1:「カス」
 2:「カス」
 3:「カス」「カス」
 4:「赤短冊」
 5:「カス」「赤短冊」
 6:「カス」「赤短冊」
 7:「カス」「カス」「赤短冊」
 8:「松に鶴(光礼)」
 9:「カス」「松に鶴(光礼)」
 10:「カス」「松に鶴(光礼)」
 11:「カス」「カス」「松に鶴(光礼)」
 12:「赤短冊」「松に鶴(光礼)」
 13:「カス」「赤短冊」「松に鶴(光礼)」
 14:「カス」「赤短冊」「松に鶴(光礼)」
 15:「カス」「カス」「赤短冊」「松に鶴(光礼)」
 1月(松)の札バターンを入力してください:7

2月の札(梅)で所持してる札の候補を次から選んで数値0～15で入力してください。
 (同一候補がある場合、候補内でどの値を入れても構いません)
 0:所持なし
 1:「カス」
 2:「カス」
 3:「カス」「カス」
 4:「赤短冊」
 5:「カス」「赤短冊」
 6:「カス」「赤短冊」
 7:「カス」「カス」「赤短冊」
 8:「梅に鶯(タネ札)」
 9:「カス」「梅に鶯(タネ札)」
 10:「カス」「梅に鶯(タネ札)」
 11:「カス」「カス」「梅に鶯(タネ札)」
 12:「赤短冊」「梅に鶯(タネ札)」
 13:「カス」「赤短冊」「梅に鶯(タネ札)」
 14:「カス」「赤短冊」「梅に鶯(タネ札)」
 15:「カス」「カス」「赤短冊」「梅に鶯(タネ札)」
 2月(梅)の札バターンを入力してください:36
 2月(梅)の札バターンを入力してください:3

3月の札(桜)で所持してる札の候補を次から選んで数値0～15で入力してください。
 (同一候補がある場合、候補内でどの値を入れても構いません)
 0:所持なし
 1:「カス」
 2:「カス」
 3:「カス」「カス」
 4:「赤短冊」
 5:「カス」「赤短冊」
 6:「カス」「赤短冊」
 7:「カス」「カス」「赤短冊」
 8:「桜に幕(光礼)」
 9:「カス」「桜に幕(光礼)」
 10:「カス」「桜に幕(光礼)」
 11:「カス」「カス」「桜に幕(光礼)」
 12:「赤短冊」「桜に幕(光礼)」
 13:「カス」「赤短冊」「桜に幕(光礼)」
 14:「カス」「赤短冊」「桜に幕(光礼)」
 15:「カス」「カス」「赤短冊」「桜に幕(光礼)」
 3月(桜)の札バターンを入力してください:8

4月の札(藤)で所持してる札の候補を次から選んで数値0～15で入力してください。
 (同一候補がある場合、候補内でどの値を入れても構いません)
 0:所持なし
 1:「カス」
 2:「カス」
 3:「カス」「カス」
 4:「短冊」
 5:「カス」「短冊」
 6:「カス」「短冊」
 7:「カス」「カス」「短冊」
 8:「藤にホトトギス(タネ札)」
 9:「カス」「藤にホトトギス(タネ札)」
 10:「カス」「カス」「藤にホトトギス(タネ札)」
 11:「カス」「カス」「藤にホトトギス(タネ札)」
 12:「短冊」「藤にホトトギス(タネ札)」
 13:「カス」「短冊」「藤にホトトギス(タネ札)」
 14:「カス」「短冊」「藤にホトトギス(タネ札)」
 15:「カス」「カス」「短冊」「藤にホトトギス(タネ札)」
 4月(藤)の札バターンを入力してください:5

5月の札(菖蒲)で所持してる札の候補を次から選んで数値0～15で入力してください。
 (同一候補がある場合、候補内でどの値を入れても構いません)
 0:所持なし
 1:「カス」
 2:「カス」
 3:「カス」「カス」
 4:「短冊」
 5:「カス」「短冊」
 6:「カス」「短冊」
 7:「カス」「カス」「短冊」
 8:「菖蒲に八つ橋(タネ札)」
 9:「カス」「菖蒲に八つ橋(タネ札)」
 10:「カス」「菖蒲に八つ橋(タネ札)」
 11:「カス」「カス」「菖蒲に八つ橋(タネ札)」
 12:「短冊」「菖蒲に八つ橋(タネ札)」
 13:「カス」「短冊」「菖蒲に八つ橋(タネ札)」
 14:「カス」「短冊」「菖蒲に八つ橋(タネ札)」
 15:「カス」「カス」「短冊」「菖蒲に八つ橋(タネ札)」
 5月(菖蒲)の札バターンを入力してください:7

6月の札(牡丹)で所持してる札の候補を次から選んで数値0～15で入力してください。
 (同一候補がある場合、候補内でどの値を入れても構いません)
 0:所持なし
 1:「カス」
 2:「カス」
 3:「カス」「カス」
 4:「青短冊」
 5:「カス」「青短冊」
 6:「カス」「青短冊」
 7:「カス」「カス」「青短冊」
 8:「牡丹に蝶(タネ札)」
 9:「カス」「牡丹に蝶(タネ札)」
 10:「カス」「牡丹に蝶(タネ札)」
 11:「カス」「カス」「牡丹に蝶(タネ札)」
 12:「青短冊」「牡丹に蝶(タネ札)」
 13:「カス」「青短冊」「牡丹に蝶(タネ札)」
 14:「カス」「青短冊」「牡丹に蝶(タネ札)」
 15:「カス」「カス」「青短冊」「牡丹に蝶(タネ札)」
 6月(牡丹)の札バターンを入力してください:10

7月の札(萩)で所持してる札の候補を次から選んで数値0～15で入力してください。
 (同一候補がある場合、候補内でどの値を入れても構いません)
 0:所持なし
 1:「カス」
 2:「カス」
 3:「カス」「カス」
 4:「短冊」
 5:「カス」「短冊」
 6:「カス」「短冊」
 7:「カス」「カス」「短冊」
 8:「萩に猪(タネ札)」
 9:「カス」「萩に猪(タネ札)」
 10:「カス」「萩に猪(タネ札)」
 11:「カス」「カス」「萩に猪(タネ札)」
 12:「短冊」「萩に猪(タネ札)」
 13:「カス」「短冊」「萩に猪(タネ札)」
 14:「カス」「短冊」「萩に猪(タネ札)」
 15:「カス」「カス」「短冊」「萩に猪(タネ札)」
 7月(萩)の札バターンを入力してください:12

8月の札(芒)で所持してる札の候補を次から選んで数値0～15で入力してください。
 (同一候補がある場合、候補内でどの値を入れても構いません)
 0:所持なし
 1:「カス」
 2:「カス」
 3:「カス」「カス」
 4:「芒に雁(タネ札)」
 5:「カス」「芒に雁(タネ札)」
 6:「カス」「芒に雁(タネ札)」
 7:「カス」「カス」「芒に雁(タネ札)」
 8:「芒に月(光礼)」
 9:「カス」「芒に月(光礼)」
 10:「カス」「芒に月(光礼)」
 11:「カス」「カス」「芒に月(光礼)」
 12:「芒に雁(タネ札)」「芒に月(光礼)」
 13:「カス」「芒に雁(タネ札)」「芒に月(光礼)」
 14:「カス」「芒に雁(タネ札)」「芒に月(光礼)」
 15:「カス」「カス」「芒に雁(タネ札)」「芒に月(光礼)」
 8月(芒)の札バターンを入力してください:9

9月の札(菊)で所持してる札の候補を次から選んで数値0～15で入力してください。
 (同一候補がある場合、候補内でどの値を入れても構いません)
 0:所持なし
 1:「カス」
 2:「カス」
 3:「カス」「カス」
 4:「青短冊」
 5:「カス」「青短冊」
 6:「カス」「青短冊」
 7:「カス」「カス」「青短冊」
 8:「菊に盃(タネ札)」
 9:「カス」「菊に盃(タネ札)」
 10:「カス」「菊に盃(タネ札)」
 11:「カス」「カス」「菊に盃(タネ札)」
 12:「青短冊」「菊に盃(タネ札)」
 13:「カス」「青短冊」「菊に盃(タネ札)」
 14:「カス」「青短冊」「菊に盃(タネ札)」
 15:「カス」「カス」「青短冊」「菊に盃(タネ札)」
 9月(菊)の札バターンを入力してください:2

10月の札(紅葉)で所持してる札の候補を次から選んで数値0～15で入力してください。
 (同一候補がある場合、候補内でどの値を入れても構いません)
 0:所持なし
 1:「カス」
 2:「カス」
 3:「カス」「カス」
 4:「青短冊」
 5:「カス」「青短冊」
 6:「カス」「青短冊」
 7:「カス」「カス」「青短冊」
 8:「紅葉に鹿(タネ札)」
 9:「カス」「紅葉に鹿(タネ札)」
 10:「カス」「紅葉に鹿(タネ札)」
 11:「カス」「カス」「紅葉に鹿(タネ札)」
 12:「青短冊」「紅葉に鹿(タネ札)」
 13:「カス」「青短冊」「紅葉に鹿(タネ札)」
 14:「カス」「青短冊」「紅葉に鹿(タネ札)」
 15:「カス」「カス」「青短冊」「紅葉に鹿(タネ札)」
 10月(紅葉)の札バターンを入力してください:1

11月の札(柳)で所持してる札の候補を次から選んで数値0～15で入力してください。
 (同一候補がある場合、候補内でどの値を入れても構いません)
 0:所持なし
 1:「カス」
 2:「短冊」
 3:「カス」「短冊」
 4:「柳に燕(タネ札)」
 5:「カス」「柳に燕(タネ札)」
 6:「短冊」「柳に燕(タネ札)」
 7:「カス」「短冊」「柳に燕(タネ札)」
 8:「小野道風に蛙(光礼)」
 9:「カス」「小野道風に蛙(光礼)」
 10:「短冊」「小野道風に蛙(光礼)」
 11:「カス」「短冊」「小野道風に蛙(光礼)」
 12:「柳に燕(タネ札)」「小野道風に蛙(光礼)」
 13:「カス」「柳に燕(タネ札)」「小野道風に蛙(光礼)」
 14:「短冊」「柳に燕(タネ札)」「小野道風に蛙(光礼)」
 15:「カス」「短冊」「柳に燕(タネ札)」「小野道風に蛙(光礼)」
 11月(柳)の札バターンを入力してください:15

12月の札(桐)で所持してる札の候補を次から選んで数値0～15で入力してください。
 (同一候補がある場合、候補内でどの値を入れても構いません)
 0:所持なし
 1:「カス」
 2:「カス」
 3:「カス」「カス」
 4:「カス」
 5:「カス」「カス」
 6:「カス」「カス」
 7:「カス」「カス」「カス」
 8:「桐に鳳凰(光礼)」
 9:「カス」「桐に鳳凰(光礼)」
 10:「カス」「桐に鳳凰(光礼)」
 11:「カス」「カス」「桐に鳳凰(光礼)」
 12:「カス」「桐に鳳凰(光礼)」
 13:「カス」「カス」「桐に鳳凰(光礼)」
 14:「カス」「カス」「桐に鳳凰(光礼)」
 15:「カス」「カス」「カス」「桐に鳳凰(光礼)」
 12月(桐)の札バターンを入力してください:8

次の札で処理します。
 1月(松)の札で所持してる札:「カス」「カス」「赤短冊」
 2月(梅)の札で所持してる札:「カス」「カス」
 3月(桜)の札で所持してる札:「カス」「赤短冊」
 4月(藤)の札で所持してる札:「カス」「短冊」
 5月(菖蒲)の札で所持してる札:「カス」「カス」「短冊」
 6月(牡丹)の札で所持してる札:「カス」「牡丹に蝶(タネ札)」
 7月(萩)の札で所持してる札:「短冊」「萩に猪(タネ札)」
 8月(芒)の札で所持してる札:「カス」「芒に月(光礼)」
 9月(菊)の札で所持してる札:「カス」
 10月(紅葉)の札で所持してる札:「カス」
 11月(柳)の札で所持してる札:「カス」「短冊」「柳に燕(タネ札)」「小野道風に蛙(光礼)」
 12月(桐)の札で所持してる札:「桐に鳳凰(光礼)」
 これでよろしいですか?
 実行する(O),再入力する(I)O

取得した札
 1月(松)の札で所持してる札:「カス」「カス」「赤短冊」
 2月(梅)の札で所持してる札:「カス」「カス」
 3月(桜)の札で所持してる札:「カス」「赤短冊」
 4月(藤)の札で所持してる札:「カス」「短冊」
 5月(菖蒲)の札で所持してる札:「カス」「カス」「短冊」
 6月(牡丹)の札で所持してる札:「カス」「牡丹に蝶(タネ札)」
 7月(萩)の札で所持してる札:「短冊」「萩に猪(タネ札)」
 8月(芒)の札で所持してる札:「カス」「芒に月(光礼)」
 9月(菊)の札で所持してる札:「カス」
 10月(紅葉)の札で所持してる札:「カス」
 11月(柳)の札で所持してる札:「カス」「短冊」「柳に燕(タネ札)」「小野道風に蛙(光礼)」
 12月(桐)の札で所持してる札:「桐に鳳凰(光礼)」

出来役
 タン(1文)
 カス(4文)

合計:5文

図2-7 実行例

※仮にすべて取得（48枚所持してる）時の役、文数は、次のようになる。

五光	10文
花見で一杯	5文
月見で一杯	5文
猪鹿蝶+タネ6	11文
赤短青短+タン4	14文
カス	15文

合計 60文×2=120文

↓ 実行例

取得した札

1月（松）の札で所持してる札：「カス」「カス」「赤短冊」「松に鶴（光札）」
2月（梅）の札で所持してる札：「カス」「カス」「赤短冊」「梅に鶯（タネ札）」
3月（桜）の札で所持してる札：「カス」「カス」「赤短冊」「桜に幕（光札）」
4月（藤）の札で所持してる札：「カス」「カス」「短冊」「藤にホトトギス（タネ札）」
5月（菖蒲）の札で所持してる札：「カス」「カス」「短冊」「菖蒲に八つ橋（タネ札）」
6月（牡丹）の札で所持してる札：「カス」「カス」「青短冊」「牡丹に蝶（タネ札）」
7月（萩）の札で所持してる札：「カス」「カス」「短冊」「萩に猪（タネ札）」
8月（芒）の札で所持してる札：「カス」「カス」「芒に雁（タネ札）」「芒に月（光札）」
9月（菊）の札で所持してる札：「カス」「カス」「青短冊」「菊に盃（タネ札）」
10月（紅葉）の札で所持してる札：「カス」「カス」「青短冊」「紅葉に鹿（タネ札）」
11月（柳）の札で所持してる札：「カス」「短冊」「柳に燕（タネ札）」「小野道風に蛙（光札）」
12月（桐）の札で所持してる札：「カス」「カス」「カス」「桐に鳳凰（光札）」

出来役

五光(10文)
花見で一杯(5文)
月見で一杯(5文)
猪鹿蝶(11文)
赤短青短(14文)
カス(15文)

合計：120文

C言語上級問題

情報

内 容 : C言語

問題形式 : (全問) プログラム記述式問題

解 答 : 問に正対するプログラムを記述し、正しい動作結果を得られたら正解になります。

難 易 度 : 上級

(標準レベルの内容とポインタ、構造体、ファイル入出力ができる程度)

問 題 数 : 2問

最終更新 : 2014年12月30日

解 答 例 : 2015年1月20日公開予定

目標時間 : 4時間

C言語上級問題	番号 氏名	
---------	----------	--

C言語上級問題

番号
氏名

※各問に対する正しいプログラムを記述しなさい。

各問題には、条件、ヒントとして条件文、図、動作結果、プログラムの例を表示しています。

実行すると動作結果通りまたは、問に正対する動作結果が得られたら正解です。

問題1

入力された金額を金種別に分類するプログラムを作成しなさい。

但し、次の条件に合うように金種計算プログラムを完成させよ。

条件：

①枚数は、考えられる数の最小になるようにすること。また今回は、一万円、五千円、千円、五百円、百円、五十円、十円、五円、一円を考慮すること。（二千円は考えない）

②プログラムは、最大10人分のデータが入るようにすること。

③プログラムでは構造体を用いてデータを作成すること、データの内訳は、図1-1のようにすること。

④一連の処理が完了したとき、「別データを入力しますか?」と問い、ユーザーがYES(1)を入力したら、入力処理から再開し、NO(0)が入力されたら終了するようにしなさい。その処理は、図1-2の処理文を利用し、main関数内に記述すること。（文中の変数名は変更しても構わない）

⑤プログラムの自作関数は、図1-3のように構成し、main関数内に記述するのは、変数宣言と自作関数、と図1-2の処理文とそれに関連する文のみとし、それ以外の入力、演算、出力は、それぞれの自作関数内で行うこと。

⑥プログラム上では次の文の記述を禁止する。

- goto構文とラベル文
- switch構文（else ifによるスイッチも含む【if、elseは可】）

⑦プログラム上では次の要素を入れること。

- ポインタ（[*][&]間接参照演算子等）
- 構造体（typedef struct ～等）
- 文字列配列

⑧出力は、画面に表示するのとtxt出力もすること。またそれぞれの出力には、それぞれ別の関数を作成すること。（呼び出しの順序は問わない）

⑨出力の際には、図1-4のように「一万円は、XXXX枚」のように表記し、体裁をそろえること。

⑩テキスト出力の際、テキストファイルには、日付、時刻データも記入すること。またそのために「time.h」をインクルードし、「tm構造体（time.hで定義されている構造体）」を使用し、「time関数、localtime関数（time.hで定義されている関数）」を使用すること。

また「time.h」を用いたプログラムの例を図1-5を参考に添付する。

C-0202

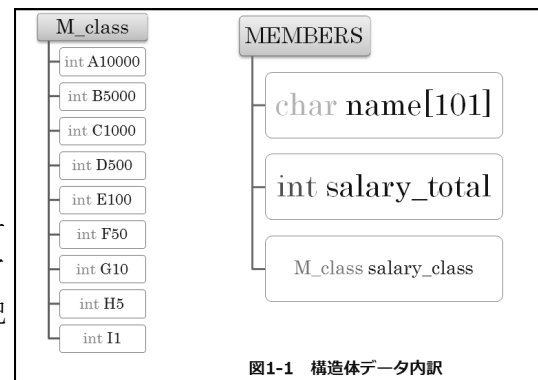


図1-1 構造体データ内訳

```
int F;
do{
    //一連の処理
    do{
        printf("別データを入力しますか?%nYes(1),NO(0)");
        scanf("%d",&F);
    }while(F!=1&&F!=0);
}while(F==1);
return 0;
```

図1-2 main関数内、終了処理の例

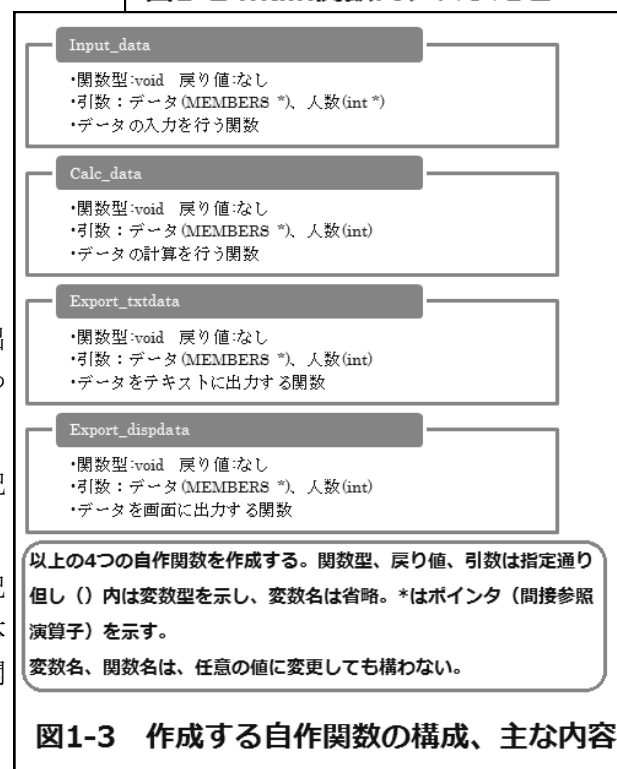


図1-3 作成する自作関数の構成、主な内容

- ①その他の各宣言（配列）の例として、図1-6のようにする。
- ②プログラムの例として、図1-7（画面表示）、図1-8（テキスト出力）に示す。
- ③プログラムは200行以内で記述すること。

```
int main(){
    time_t timeValue;
    struct tm *timeObject;

    time(&timeValue);
    timeObject = localtime(&timeValue);
    printf("ただいまの日付、時刻は、%d年%d月%d日 %d時%d分%d秒です。",
        timeObject->tm_year + 1900, timeObject->tm_mon + 1, timeObject->tm_mday,
        timeObject->tm_hour, timeObject->tm_min, timeObject->tm_sec);
    return 0;
}
```

注) time.hをインクルードすること

図1-5 time.hを用いた現在時刻取得の例

一万円は、	12枚
五千円は、	0枚
千円は、	3枚
五百円は、	0枚
百円は、	4枚
五十円は、	1枚
十円は、	0枚
五円は、	1枚
一円は、	1枚

図1-4 出力例

給料金種別計算機
 入力する人数を指定してください(最大10人):2
 1人名前入力【半角100文字(全角50文字)以内、空白入力禁止】:羽鶴太郎
 羽鶴太郎さんの給料入力【0以上100,000,000以下】:205758
 2人名前入力【半角100文字(全角50文字)以内、空白入力禁止】:希条裕樹
 希条裕樹さんの給料入力【0以上100,000,000以下】:211835
 給料金種内訳明細書

1人目:羽鶴太郎さんの給料205758円の金種内訳

一万円は、	20枚
五千円は、	1枚
千円は、	0枚
五百円は、	1枚
百円は、	2枚
五十円は、	1枚
十円は、	0枚
五円は、	1枚
一円は、	3枚

2人目:希条裕樹さんの給料211835円の金種内訳

一万円は、	21枚
五千円は、	0枚
千円は、	1枚
五百円は、	1枚
百円は、	3枚
五十円は、	0枚
十円は、	3枚
五円は、	1枚
一円は、	0枚

-金種別累計枚数-

一万円は、	41枚
五千円は、	1枚
千円は、	1枚
五百円は、	2枚
百円は、	5枚
五十円は、	1枚
十円は、	3枚
五円は、	2枚
一円は、	3枚

別データを入力しますか?
 Yes(1),NO(0)

この例は、2人のデータの入力を行い、その出力結果(画面)を表示した例である。
 金種別累計枚数は、2人の各金種の合計である。
 別データを入力しますか?の質問に対し、0ならそのまま終了、1なら処理を最初から行うようにする。(この画面では、その入力前の状態を表しています、このとき既に入力したデータは、テキストデータに出力されている)

図1-7 画面出力例

```
const int
MONEY[]={10000,5000,1000,500,100,50,10,5,1};

【演算するための金種の宣言】

static char
*Money_Ord[]={ "一万円","五千円","千円","五百円","百円","五十円","十円","五円","一円"};

【出力時の文字列】
```

図1-6 関数外で宣言する各宣言（配列）の例

給料金種内訳明細書【2014年12月26日 3時27分2秒生成】.txt
 給料金種内訳明細書↓

-----↓

1人目:羽鶴太郎さんの給料205758円の金種内訳↓

一万円は、	20枚↓
五千円は、	1枚↓
千円は、	0枚↓
五百円は、	1枚↓
百円は、	2枚↓
五十円は、	1枚↓
十円は、	0枚↓
五円は、	1枚↓
一円は、	3枚↓

2人目:希条裕樹さんの給料211835円の金種内訳↓

一万円は、	21枚↓
五千円は、	0枚↓
千円は、	1枚↓
五百円は、	1枚↓
百円は、	3枚↓
五十円は、	0枚↓
十円は、	3枚↓
五円は、	1枚↓
一円は、	0枚↓

-----↓

-金種別累計枚数-↓

一万円は、	41枚↓
五千円は、	1枚↓
千円は、	1枚↓
五百円は、	2枚↓
百円は、	5枚↓
五十円は、	1枚↓
十円は、	3枚↓
五円は、	2枚↓
一円は、	3枚↓

-----↓

[EOF]

ファイル名:給料金種内訳明細書【2014年12月26日 3時27分2秒生成】.txt

図1-8 テキストファイル出力例

問題2

こいこい役判定、文数計算プログラムを作成せよ。
但し、次のルール、条件に合うようにプログラムを完成させよ。

こいこい役判定、文数についてのルール

※図2-1にも絵付で説明しています。

①判定する役と文数は、次の13種類とする。

(1)、五光 (10文)

・光札5枚

(2)、四光 (8文)

・光札4枚但し「小野道風に蛙」を含まない

(3)、雨四光 (7文)

・光札4枚但し「小野道風に蛙」を含む

(4)、三光 (5文)

・光札3枚但し「小野道風に蛙」を含まない

(5)、花見で一杯 (5文)

・「菊に盃」「桜に幕」

(6)、月見で一杯 (5文)

・「菊に盃」「芒に月」

(7)、猪鹿蝶 (5文+猪鹿蝶以外のタネ札の数)

・「萩に猪」「紅葉に鹿」「牡丹に蝶」

(8)、赤短青短 (10文+赤短、青短以外の短冊札の数)

・赤短札3枚と青短札3枚

(9)、赤短 (5文+赤短以外の短冊札の数)

・赤短札3枚

(10)、青短 (5文+青短以外の短冊札の数)

・青短札3枚

(11)、タネ (1文)

・タネ札(10点札) 5枚以上、6枚目から1枚毎に+1文

(12)、タン (1文)

・短冊札5枚以上、6枚目から1枚毎に+1文

(13)、カス (1文)

・カス札10枚以上、11枚目から1枚毎に+1文

②①以外の役、手四、くっつきの判定は行わない。

③役ルールとして次の条件を考慮すること。

・(7)猪鹿蝶が成立した時点で(11)タネ消滅

・(8)赤短青短が成立した時点で(9)赤短、(10)青短、(12)タン消滅

・(9)赤短が成立した時点で(12)タン消滅

・(10)青短が成立した時点で(12)タン消滅

④札のルールについて次のように定義する。

・光札：「鶴に松」「桜に幕」「芒に月」「小野道風に蛙」「桐に鳳凰」

・タネ札(10点札)：「梅に鶯」「藤にほととぎす」「菖蒲に八ツ橋」「牡丹に蝶」「萩に猪」
「芒に雁」「菊に杯」「紅葉に鹿」「柳に燕」

・短冊札：芒(8月)、桐(12月)以外に1枚ずつ計10枚

- ・カス札：上記以外
- ・「小野道風に蛙」は、カス札、タネ札に計上される。

※詳細は、図2-1参照

⑤各月毎の花鳥風月を次のように定義する。

- 1月：松
- 2月：梅
- 3月：桜
- 4月：藤
- 5月：菖蒲（アヤメ）
- 6月：牡丹（ボタン）
- 7月：萩
- 8月：芒（ススキ）
- 9月：菊
- 10月：紅葉（モミジ）
- 11月：柳
- 12月：桐

⑥合計7文以上の場合は、文数は2倍になる。

⑦こいこいルールは次のURLを参考に定義するものとする。

http://www.nintendo.co.jp/n09/hana-kabu_games/

札について	役について
1月（松） <small>松に鶴</small>	五光 10文 <small>光札5枚</small>
2月（梅） <small>梅に鶯</small>	四光 8文 <small>小野道風に蛙（柳に雨） 以外の光札4枚</small>
3月（桜） <small>桜に鶯</small>	雨四光 7文 <small>小野道風に蛙（柳に雨）含む光札4枚</small>
4月（藤） <small>藤に不如帰</small>	三光 5文 <small>小野道風に蛙（柳に雨） 以外の光札3枚</small>
5月（菖蒲） <small>菖蒲に八つ橋</small>	花見で一杯 5文
6月（牡丹） <small>牡丹に蝶</small>	月見で一杯 5文
7月（萩） <small>萩に猪</small>	猪鹿蝶 5文 <small>猪鹿蝶以外のタネ札1枚につき+1文</small>
8月（芒） <small>芒に月、芒に雁</small>	赤短青短 10文 <small>赤短青短以外の短冊札1枚につき+1文</small>
9月（菊） <small>菊に盃</small>	赤短 5文 <small>赤短以外の短冊札1枚につき+1文</small>
10月（紅葉） <small>紅葉に鹿</small>	青短 5文 <small>青短以外の短冊札1枚につき+1文</small>
11月（柳） <small>小野道風に蛙、柳に燕</small>	タネ 1文 <small>小野道風に蛙も タネ札に計上する</small> <small>タネ札5枚以上で成立 6枚以上は1枚につき+1文</small>
12月（桐） <small>桐に風扇</small>	<small>のうち5枚以上、6枚目からは1枚増える毎に+1文</small>
<small>赤字…光札 緑字…タネ札</small>	タン 1文 <small>短冊札5枚以上で成立 6枚以上は1枚につき+1文</small> <small>のうち5枚以上、6枚目からは1枚増える毎に+1文</small>
図2-1こいこいの札、 役について	カス 1文 <small>カス札10枚以上で成立 11枚以上は、 1枚につき+1文</small> <small>小野道風に蛙も カス札に計上する</small> <small>のうち10枚以上、11枚目からは1枚増える毎に+1文</small>

条件：

①役と文数、札については、全て上記ルールに従うものとする。

②プログラムでは配列（文字列配列含む）、列挙体を用いてデータを作成すること、但し列挙体は、プログラムを明確にするために使用する、データの内訳は、図2-2のようにすること。

③プログラムの処理構成は、入力、計算、出力共すべて自作関数で行うものとする。

④プログラム上では次の文の記述を禁止する。

- ・ goto構文とラベル文
- ・ switch構文 (else if、if、elseは可)

⑤プログラム上では次の要素を入れること。

- ・ ポインター ([*][&]間接参照演算子等)
- ・ 列挙体 (enum)
- ・ 文字列配列

⑥main関数に記述するのは、タイトル「こいこい役計算機」とデータの宣言と自作関数のみとする。

⑦入力処理の関数

(Input_data) は、各月毎に取得したパターン (16通り) を明示し、数字0~15で解答させる。12回行い、データを16進数で保存するようにする。但、全てのパターンは、281,474,976,710,656通り (=6byte) でint型では最大2,147,483,647bit (=4byte-符号ビット (1bit)) の値しか保存できないので、パターンは、(unsigned) long long型 (=8byte) を使うこと。メモリの仕様を図2-3で示す。(解答例ではunsigned long long型で記述しています。)

また、0~15以外の数字が入力されたら、再入力させること。

⑧入力処理が終了したら、取得したパターンを基に、取得した札を明示したうえで、このパターンで処理を実行するか尋ねる事。

⑨計算処理を行う関数 (Calc_Fuda) では、札種別の枚数 [13]、48枚の所持の有無 [48]、役別文数 [13] (全て配列) の値を計算すること。

```
static char *KITFG[]={松,梅,桜,藤,菖蒲,牡丹,萩,芒,菊,紅葉,柳,桐};//花鳥風月(月馳)
static char *FUDA[]={//札の文字
"カス","カス","赤短冊","松に鶴(光札)","//1月(松)
"カス","カス","赤短冊","梅に鶯(タネ札)","//2月(梅)
"カス","カス","赤短冊","桜に幕(光札)","//3月(桜)
"カス","カス","短冊","藤にホトギス(タネ札)","//4月(藤)
"カス","カス","短冊","菖蒲に八つ橋(タネ札)","//5月(菖蒲)
"カス","カス","青短冊","牡丹に蝶(タネ札)","//6月(牡丹)
"カス","カス","短冊","萩に猪(タネ札)","//7月(萩)
"カス","カス","芒に雁(タネ札)","芒に月(光札)","//8月(芒)
"カス","カス","青短冊","菊に盃(タネ札)","//9月(菊)
"カス","カス","青短冊","紅葉に鹿(タネ札)","//10月(紅葉)
"カス","短冊","柳に燕(タネ札)","小野道風に蛙(光札)","//11月(柳)
"カス","カス","カス","柳に鳳凰(光札)","//12月(桐)
};
static char *N_YAKU[]={//役の名前と探査、表示順序
"五光","四光","雨四光","三光","花見で一杯","月見で一杯","猪鹿蝶","赤短青短","赤短","青短","タネ","タン","カス"
};
static int YAKU_N[]={//役の文数
10,8,7,5,5,5,5,10,5,5,1,1,1
};
typedef enum{//役の名前と探査、表示順序
NAME_YAKU;
};
typedef enum{//札の名前
カス札=1,タン札,赤短札,青短札,タネ札,猪鹿蝶札,盃札,松に鶴,桜に幕,芒に月,小野道風に蛙,柳に鳳凰
NAME_FUDA;
};
static int FUDA_NUM[]={//札の種類
1,1,1,3,8,7,1,1,1,1,1,3,9,7,3,1,1,1,2,5,1,1,4,8,7,7,1,1,1,5,10,7,9,1,1,4,7,10,1,1,4,8,11,11,1,1,1,1,1,12
};
/*
カス札は、全て:1 count
短冊札は、無地:2,赤短:3,青短:4 count
タネ札は、梅に鶯:5,藤に不知火:5,菖蒲に八つ橋:5,牡丹に蝶:5,萩に猪:6,芒に雁:5,菊に盃:7,紅葉に鹿:8,柳に燕:5,光札は、松に鶴:8,桜に幕:9,芒に月:10,小野道風に蛙:11,柳に鳳凰:12
*/
```

図2-2データの構造 (宣言例)

変数サイズ一覧		1bit	1byte	データ部	符号ビット
int型(16bit型)	2byte(16bit)	データ容量: 15bit (-32,768~32,767)			
int型(32bit型)	4byte(32bit)	データ容量: 31bit (-2,147,483,648~2,147,483,647)			
long型(32bit型)	4byte(32bit)	データ容量: 31bit (-2,147,483,648~2,147,483,647)			
long long型(32bit型)	8byte(64bit)	データ容量: 63bit (-9,223,372,036,854,775,808~9,223,372,036,854,775,807)			
unsigned long long型(32bit型)	8byte(64bit)	データ容量: 64bit (0~18,446,744,073,709,551,615)			
int型(64bit型)	4byte(32bit)	データ容量: 31bit (-2,147,483,648~2,147,483,647)			
long型(64bit型)	8byte(64bit)	データ容量: 63bit (-9,223,372,036,854,775,808~9,223,372,036,854,775,807)			
long long型(64bit型)	8byte(64bit)	データ容量: 63bit (-9,223,372,036,854,775,808~9,223,372,036,854,775,807)			
unsigned long long型(64bit型)	8byte(64bit)	データ容量: 64bit (0~18,446,744,073,709,551,615)			
今回必要なパターン位置	6byte(48bit)	12月11月10月9月8月7月6月5月4月3月2月1月			
パターン保存要件	6byte(48bit)	データ容量: 48bit (0~281,474,976,710,655) (花札の全取得パターン数 281,474,976,710,656通り)			

図2-3 メモリの仕様

⑩出力処理 (Output_data) では、計算処理で計算した、役別文数を使い、出力を行う、またその際所持している札も明示すること。

⑪所持している札を明示する際、月ごとに表示し、その時その月の花鳥風月、札の名前を明示すること。明示は、入力の最後の確認、出来役を表示するときに行う。

明示の例を図2-4で示す。

⑫関数の定義を図2-5で示す。

⑬データを格納する配列の例を図2-2で示す。

⑭図2-2データの定義、図2-5関数の定義で使用した名前 (関数名、配列名、列挙体名及び列挙体で定義される文字) は、変更しても構わない。

⑮図では、列挙体の文字は明確にするために日本語で表記していますが、コンパイラによっては日本語表記ができない場合があるので、その場合は、ローマ字や英語表記で記述しても構わない。

⑯出力結果の例を図2-7で示す。

⑰プログラムは、300行以内で記述すること。

```

1月 (松) の札で所持してる札: 「カス」 「カス」 「赤短冊」
2月 (梅) の札で所持してる札: 「カス」 「カス」
3月 (桜) の札で所持してる札: 「カス」 「赤短冊」
4月 (藤) の札で所持してる札: 「カス」 「短冊」
5月 (菖蒲) の札で所持してる札: 「カス」 「カス」 「短冊」
6月 (牡丹) の札で所持してる札: 「カス」 「牡丹に蝶 (タネ札)」
7月 (萩) の札で所持してる札: 「短冊」 「萩に猪 (タネ札)」
8月 (芒) の札で所持してる札: 「カス」 「芒に月 (光札)」
9月 (菊) の札で所持してる札: 「カス」
10月 (紅葉) の札で所持してる札: 「カス」
11月 (柳) の札で所持してる札: 「カス」 「短冊」 「柳に燕 (タネ札)」 「小野道風に蛙 (光札)」
12月 (桐) の札で所持してる札: 「桐に鳳凰 (光札)」

```

図2-4 所持札表示の例

```

Input_data
・関数型: void 戻り値なし
・引数: パターンデータ (unsigned long long *)
・データの入力を行う関数

Calc_Fuda
・関数型: void 戻り値なし
・引数: パターンデータ (unsigned long long)、所持札の有無 (1/0) を入れる48の配列 (int *)、役別の文数を入れる13の配列 (int *)、札種別枚数を入れる13の配列 (int *)
・パターンデータから、所持札の有無、札種別枚数、役の有無とその文数の計算を行う関数

Show_Fuda
・関数型: void 戻り値なし
・引数: パターンデータ (unsigned long long)
・パターンデータより取得札を画面に表示する関数 (図2-4)

Output_data
・関数型: void 戻り値なし
・引数: データ (MEMBERS*), 人数 (int)
・データを画面に出力する関数

```

以上の4つの自作関数を作成する。関数型、戻り値、引数は指定通り但し () 内は変数型を示し、変数名は省略。*はポインタ (間接参照演算子) を示す。
変数名、関数名は、任意の値に変更しても構わない。

図2-5 作成する自作関数の構成、主な内容

※図2-6「実行例 (簡略版)」は、ここでは省略する。

※仮にすべて取得 (48枚所持してる) 時の役、文数は、次のようになる。

五光	10文
花見で一杯	5文
月見で一杯	5文
猪鹿蝶+タネ6	11文
赤短青短+タン4	14文
カス	15文
<hr/>	
合計	60文 × 2 = 120文

```

-----
取得した札
1月 (松) の札で所持してる札: 「カス」 「カス」 「赤短冊」 「松に鶴 (光札)」
2月 (梅) の札で所持してる札: 「カス」 「カス」 「赤短冊」 「梅に鶯 (タネ札)」
3月 (桜) の札で所持してる札: 「カス」 「カス」 「赤短冊」 「桜に幕 (光札)」
4月 (藤) の札で所持してる札: 「カス」 「カス」 「短冊」 「藤にホトトギス (タネ札)」
5月 (菖蒲) の札で所持してる札: 「カス」 「カス」 「短冊」 「菖蒲に八つ橋 (タネ札)」
6月 (牡丹) の札で所持してる札: 「カス」 「カス」 「青短冊」 「牡丹に蝶 (タネ札)」
7月 (萩) の札で所持してる札: 「カス」 「カス」 「短冊」 「萩に猪 (タネ札)」
8月 (芒) の札で所持してる札: 「カス」 「カス」 「芒に雁 (タネ札)」 「芒に月 (光札)」
9月 (菊) の札で所持してる札: 「カス」 「カス」 「青短冊」 「菊に盃 (タネ札)」
10月 (紅葉) の札で所持してる札: 「カス」 「カス」 「青短冊」 「紅葉に鹿 (タネ札)」
11月 (柳) の札で所持してる札: 「カス」 「短冊」 「柳に燕 (タネ札)」 「小野道風に蛙 (光札)」
12月 (桐) の札で所持してる札: 「カス」 「カス」 「カス」 「桐に鳳凰 (光札)」
-----
出来役
  五光 (10文)
  花見で一杯 (5文)
  月見で一杯 (5文)
  猪鹿蝶 (11文)
  赤短青短 (14文)
  カス (15文)
-----
合計: 120文

```


解答例について

- ①解答は、一例として掲載しています。
- ②問題で指定されているもの以外の、変数名（配列、構造体名、列挙体名含む）、関数名および文字列配列や、列挙体宣言、`define`宣言等で定義した名前と文字列は、変更しても構わない。
- ③問題文中の条件をすべて満たしていれば正解とする。
- ④この解答は、「Microsoft Visual C++ 2010」にて作成、動作させており、問題文中の図の一部（ソースコード）、解答例のソースコードはこれで示しています。
- ⑤解答例のソースコードは、「Microsoft Visual C++ 2010」ですべて動作確認済み、問題文中の実行例の図の通り（入力によって変化します）になります。
- ⑥解答例は、ホームページ上にも公開しています。サイト内では、カラーで公開しています。
- ⑦【問題2について】列挙体の定義の中に日本語（2バイト文字）を使用しています。一部のコンパイラでは、エラーを出力する場合があります。
- ⑧【¥について】「¥」の表記は、日本語版及び日本語対応のコンパイラで表示されます。コンパイラによっては、「\」で表記される場合があります。
- ⑨問題文中の図の著作権は、`indirect`に帰属する。

次ページより解答例を示す。

問題1

```
1 #include<stdio.h>
2 #include<time.h>
3 const int MONEY[]={10000,5000,1000,500,100,50,10,5,1}; //2千円を含めない配列
4 static char *Money_Ord[]={“一万円”,“五千円”,“千円”,“五百円”,“百円”,“五十円”,“十円”,“五円”,“一円”,};
5 typedef struct {
6     int A10000;
7     int B5000;
8     int C1000;
9     int D500;
10    int E100;
11    int F50;
12    int G10;
13    int H5;
14    int I1;
15 }M_class;
16
17 typedef struct {
18     char name[104];
19     int salary_total;
20     M_class salary_class;
21 }MEMBERS;
22
23 void Input_data(MEMBERS *a,int *member){
24     printf(“給料金種別計算機\n”);
25     do{
26         printf(“入力する人数を指定してください(最大10人):”);
27         scanf(“%d”,member);
28     }while(*member<1||*member>10);
29     for(int i=0;i<*member;i++){
30         printf(“%d人目名前入力【半角100文字(全角50文字)以内、空白入力禁止】:”,i+1);
31         scanf(“%100s”,a[i].name);
32         do{
33             printf(“%sさんの給料入力【0以上100,000,000以下】:”,a[i].name);
34             scanf(“%d",&a[i].salary_total);
35             if(a[i].salary_total<0||a[i].salary_total>100000000)printf(“!再入力してください!”);
36         }while(a[i].salary_total<0||a[i].salary_total>100000000);
37     }
38 }
39
40 void Export_txtdata(MEMBERS *a,int member){
41     FILE *FP;
42     time_t timeValue;
43     struct tm *timeObject;
44     char name[100];
45     int SUM[9]={0},*AS;
46     time(&timeValue);
47     timeObject = localtime(&timeValue);
48     sprintf(name,“給料金種内訳明細書【%d年%d月%d日 %d時%d分%d秒生成】.txt”,timeObject->tm_year + 1900,timeObject->tm_mon + 1,timeObject->tm_mday,
49             ,timeObject->tm_hour,timeObject->tm_min,timeObject->tm_sec);
50     FP=fopen(name,“w”);
51     fprintf(FP,“%s\n”,name);
52     fprintf(FP,“給料金種内訳明細書\n-----\n”);
53     for(int j=0;j<member;j++){
54         fprintf(FP,“%2d人目: %sさんの給料%6d円の金種内訳\n”,j+1,a[j].name,a[j].salary_total);
55         AS=&a[j].salary_class.A10000;
56         for(int i=0;i<9;i++){
57             SUM[i]+=*AS;
58             fprintf(FP,“%sは、%6d枚\n”,Money_Ord[i],*AS);
59             AS++;
60         }
61     }
62     fprintf(FP,“-----\n”);
63     fprintf(FP,“金種別累計枚数\n”);
64     fprintf(FP,“-----\n”);
65     for(int i=0;i<9;i++)fprintf(FP,“%sは、%6d枚\n”,Money_Ord[i],SUM[i]);
66     fprintf(FP,“-----\n”);
67     fclose(FP);
68 }
69
70
71
72 void Export_dispdata(MEMBERS *a,int member){
73     int SUM[9]={0},*AS;
74     printf(“給料金種内訳明細書\n-----\n”);
75     for(int j=0;j<member;j++){
76         printf(“%2d人目: %sさんの給料%6d円の金種内訳\n”,j+1,a[j].name,a[j].salary_total);
77         AS=&a[j].salary_class.A10000;
78         for(int i=0;i<9;i++){
79             SUM[i]+=*AS;
80             printf(“%sは、%6d枚\n”,Money_Ord[i],*AS);
81             AS++;
82         }
83     }
84     printf(“-----\n”);
85     printf(“金種別累計枚数\n”);
86     printf(“-----\n”);
87     for(int i=0;i<9;i++)printf(“%sは、%6d枚\n”,Money_Ord[i],SUM[i]);
88     printf(“-----\n”);
89 }
90
91 void Calc_data(MEMBERS *a,int member){
92     int s,n,*P;
93     for(int j=0;j<member;j++){
94         n=a[j].salary_total;
95         P=&a[j].salary_class.A10000;
96         for(int i=0;i<9;i++){
97             s=n/MONEY[i];
98             n=s*MONEY[i];
99             *P=s;
100            P++;
101        }
102        *P=n;
103    }
104 }
105
106 int main(void){
107     int N=0,F=0;
108     MEMBERS ME[10];
109     do{
110         Input_data(ME,&N);
111         Calc_data(ME,N);
112         Export_txtdata(ME,N);
113         Export_dispdata(ME,N);
114         do{
115             printf(“別データを入力しますか? %nYes(1),NO(0)”);
116             scanf(“%d",&F);
117         }while(F!=1&&F!=0);
118     }while(F==1);
119     return 0;
120 }
```

問題2

```
1 #include<stdio.h>
2 static char *KTFG[]={“松”,“桜”,“藤”,“菖蒲”,“牡丹”,“萩”,“芒”,“菊”,“紅葉”,“柳”}; //花鳥風月 (月期)
3 static char *FUUDA[]={“礼の文字”};
4 “カス”,“カス”,“赤短冊”, “松に鶴 (光札)”, “ //1月 (松)
5 “カス”,“カス”,“赤短冊”, “柳に鶯 (タネ札)”, “ //2月 (柳)
6 “カス”,“カス”,“赤短冊”, “桜に鶯 (光札)”, “ //3月 (桜)
7 “カス”,“カス”,“短冊”, “藤にホトトギス (タネ札)”, “ //4月 (藤)
8 “カス”,“カス”,“短冊”, “菖蒲に八つ橋 (タネ札)”, “ //5月 (菖蒲)
9 “カス”,“カス”,“青短冊”, “牡丹に蝶 (タネ札)”, “ //6月 (牡丹)
10 “カス”,“カス”,“短冊”, “萩に鶯 (タネ札)”, “ //7月 (萩)
11 “カス”,“カス”,“芒に雁 (タネ札)”, “芒に月 (光札)”, “ //8月 (芒)
12 “カス”,“カス”,“青短冊”, “菊に盃 (タネ札)”, “ //9月 (菊)
13 “カス”,“カス”,“青短冊”, “紅葉に鹿 (タネ札)”, “ //10月 (紅葉)
14 “カス”,“短冊”, “小野道風に蛙 (タネ札)”, “ //11月 (柳)
15 “カス”,“カス”,“カス”, “柳に鳳凰 (光札)”, “ //12月 (柳)
16 };
17
18 static char *N_YAKU[]={“役の名前と探査、表示順序”};
19 “五光”,“四光”,“雨四光”,“三光”,“花見で一杯”,“月見で一杯”,“猪鹿蝶”,“赤短青短”,“赤短”,“青短”,“タネ”,“タン”,“カス”
20 };
21
22 static int YAKU_N[]={“役の文数”};
23 “10,8,7,5,5,5,5,10,5,5,1,1,1”
24 };
25
26 #typedef enum {“役の名前と探査、表示順序”};
27 “五光,四光,雨四光,三光,花見で一杯,月見で一杯,猪鹿蝶,赤短青短,赤短,青短,タネ,タン,カス”
28 #NAME_YAKU;
29
30 #typedef enum {“礼の名前”};
31 “カス礼=1,タン礼,赤短札,青短札,タネ札,猪鹿蝶礼,盃礼,松に鶴,桜に鶯,芒に月,小野道風に蛙,柳に鳳凰”
32 #NAME_FUUDA;
33
34 static int FUUDA_NUM[]={“礼の種類”};
35 “1,1,1,3,8, //1月
36 “1,1,3,5, //2月
37 “1,1,3,3, //3月
38 “1,1,2,5, //4月
39 “1,1,2,5, //5月
40 “1,1,4,6, //6月
41 “1,1,5,6, //7月
42 “1,1,5,10, //8月
43 “1,1,4,7, //9月
44 “1,1,4,6, //10月
45 “1,2,5,11, //11月
46 “1,1,1,12 //12月
47 };
48
49 /*
50 カス礼は、全て:1 count
51 短冊礼は、無地:2、赤短:3、青短:4 count
52 タネ札は、柳に鶯:5、藤に不知火:5、菖蒲に八つ橋:5、牡丹に蝶:6、萩に鶯:6、芒に雁:5、菊に盃:7、紅葉に鹿:6、柳に燕:5、
53 光札は、松に鶴:6、桜に鶯:6、芒に月:10、小野道風に蛙:11、柳に鳳凰:12
54
55
56 void Input_data(unsigned long long *);
57 void Show_Fuda(unsigned long long);
58 void Calc_Fuda(unsigned long long,int * ,int * ,int *);
59 void Output_data(unsigned long long,int *);
60
61
62 #void Input_data(unsigned long long *pattern){
63 int b,f;
64 do{
65 printf(“とった札を月ごとに0~15の数値で入力してください。\\n”);
66 for(int i=0;i<12;i++){
67 printf(“%d月の札(%s)で所持してる札の種類を次から選んで数値0~15で入力してください。\\n (同一候補がある場合、候補間での値を入れてもらっても構いません) \\n”
68 ,i+1,KTFG[i]);
69 for(int j=0;j<16;j++){
70 printf(“%d:”);
71 if(j==0){
72 printf(“所持なし\\n”);
73 continue;
74 }
75 for(int k=0;k<4;k++){if((j>k)&&k%2==1)printf(“ [%s]”,FUUDA[i+k]);} //ずらす数
76 printf(“\\n”);
77 }
78 do{
79 printf(“%d月 (%s) の札パターンを入力してください。”,i+1,KTFG[i]);
80 scanf(“%d”,&b);
81 while(b<0|b>15);
82 *pattern+=(unsigned long long)b<<(i*4);
83 printf(“-----\\n”);
84 }
85 printf(“次の札で処理します。\\n”);
86 Show_Fuda(*pattern);
87 do{
88 printf(“これでよろしいですか?\\n実行する(0),再入力する(1)”);
89 scanf(“%d”,&f);
90 while(f!=0&&f!=1);
91 printf(“-----\\n”);
92 }while(f==1);
93 }
94
95 #void Calc_Fuda(unsigned long long pattern,int *Have,int *Yaku,int *Y){
96 int Hkari=0;
97 for(int i=0;i<12;i++){
98 for(int j=0;j<4;j++){
99 (pattern>>((i*4+j))%2==1 ? *Have=1 : *Have=0;
100 }
101 }
102 for(int i=0;i<48;i++)Have--;
103 for(int i=0;i<48;i++){
104 for(int j=1;j<13;j++){
105 if(FUUDA_NUM[j]==*Have+1)Y[j]++; //カス、タン (無地)、赤短、青短、タネ (右以外)、猪鹿蝶礼、盃礼、松に鶴、桜に鶯、芒に月、小野道風に蛙、柳に鳳凰
106 if(FUUDA_NUM[j]==0&&*Have==1)Hkari++; //光札カウント
107 if(FUUDA_NUM[j]==小野道風に蛙&&*Have==1)Y[カス礼]++,Y[タネ札]++; //雨札補正
108 }
109 }
110 //役判定、文数補正
111 if(Y[カス礼]==10)Yaku[カス]=Y[カス礼]-10; //カス
112 if(Y[タン礼]+Y[猪鹿蝶礼]==5)Yaku[タン]=Y[タン礼]+Y[猪鹿蝶礼]+Y[赤短札]-5; //タン
113 if(Y[タネ札]+Y[猪鹿蝶礼]+Y[盃礼]==5)Yaku[タネ]=Y[タネ札]+Y[猪鹿蝶礼]+Y[盃礼]-5; //タネ
114
115 if(Hkari==5) Yaku[五光]=YAKU_N[五光]; //五光
116 if(Hkari==4&&Y[小野道風に蛙]==1) Yaku[四光]=YAKU_N[四光]; //四光
117 if(Hkari==4&&Y[小野道風に蛙]==1) Yaku[雨四光]=YAKU_N[雨四光]; //雨四光
118 if(Hkari==3&&Y[小野道風に蛙]==1) Yaku[三光]=YAKU_N[三光]; //三光
119
120 if(Y[桜に鶯]==1&&Y[盃礼]==1)Yaku[花見で一杯]=YAKU_N[花見で一杯]; //花見で一杯
121 if(Y[芒に雁]==1&&Y[盃礼]==1)Yaku[月見で一杯]=YAKU_N[月見で一杯]; //月見で一杯
122 if(Y[猪鹿蝶礼]==3)Yaku[猪鹿蝶]=YAKU_N[猪鹿蝶]+Y[タネ札],Yaku[タネ]=0; //猪鹿蝶
123
124 if(Y[赤短札]==3&&Y[青短札]==3)Yaku[赤短青短]=YAKU_N[赤短青短]+Y[タン礼],Yaku[タン]=0; //赤短青短
125 else if(Y[赤短札]==3)Yaku[赤短]=YAKU_N[赤短]+Y[タン礼]+Y[青短札],Yaku[タン]=0; //赤短
126 else if(Y[青短札]==3)Yaku[青短]=YAKU_N[青短]+Y[タン礼]+Y[赤短札],Yaku[タン]=0; //青短
127 }
128
129 #void Show_Fuda(unsigned long long pattern){
130 int b;
131 for(int i=0;i<12;i++){
132 b=(unsigned long long)(pattern>>i*4)%16;
133 printf(“%d月 (%s) の札で所持してる札:”,i+1,KTFG[i]);
134 if(b==0){
135 printf(“所持なし\\n”);
136 continue;
137 }
138 for(int k=0;k<4;k++){if((b>k)&&k%2==1)printf(“ [%s]”,FUUDA[i+k]);} //ずらす数
139 printf(“\\n”);
140 }
141 }
142
143 #void Output_data(unsigned long long pattern,int *Yaku){
144 int Mon=0;
145
146 for(int i=0;i<13;i++){if(Yaku[i]>0)Mon+=Yaku[i];
147 printf(“取得した札\\n”);
148 Show_Fuda(pattern);
149 printf(“-----\\n”);
150 if(Mon==0){
151 printf(“出役\\n”);
152 for(int i=0;i<13;i++){if(Yaku[i]>0)printf(“%1s(%2d文)\\n”,N_YAKU[i],Yaku[i]);
153 if(Mon==7)Mon+=2;
154 }
155 printf(“-----\\n合計: %2d文\\n,Mon);
156 }
157 else printf(“*出来役はありません*”);
158 }
159
160 #int main(void){
161 unsigned long long pattern=0; //所持してる札 (16進データ)
162 int H_Fuda[48]={0}; //所持している札「配列」 (1/0)
163 int Yaku_Mon[13]={0}; //役別文数 (一三種類)
164 int Y_Fuda[13]={0}; //礼種類別所持数 (一三種類)
165 printf(“こいこい役計書機\\n”);
166 Input_data(&pattern);
167 Calc_Fuda(pattern,H_Fuda,Yaku_Mon,Y_Fuda);
168 Output_data(pattern,Yaku_Mon);
169 }
```


2014年12月31日 第一版 第一刷 発行

著 者 indirect

発行者 indirect

発行所 何となく作ってみたサイト

URL <http://indirect2435.jimdo.com/>