

プログラミング基礎(C) 挑戦課題

提出期限： 7月9日(月)

発表： 7月16日(月)

【問題1】

現在の時刻における世界の主要国や都市の時刻を出力するプログラムをつくりなさい。構造体配列を使用します。ただし、日本との時差は次のようになっています。

時 差	国または都市
+2	ニューカレドニア
+1	サイパン
-1	フィリピン, 香港, 台湾, マレーシア, シンガポール
-2	タイ
-4	インド
-6	イラク, サウジアラビア, ソマリア, マダガスカル
-7	ギリシャ, エジプト, スーダン
-8	イタリア, スイス, ドイツ
-14	ニューヨーク, ペルー
-17	ロサンゼルス
-19	ハワイ

《ヒント》 時刻を知るには、ヘッダファイル `time.h` をインクルードして、`time` 関数と `ctime` 関数あるいは `localtime` 関数を使用する。

▼出力例

*** 選択メニュー ***

現在の時刻 11:06 (JST)

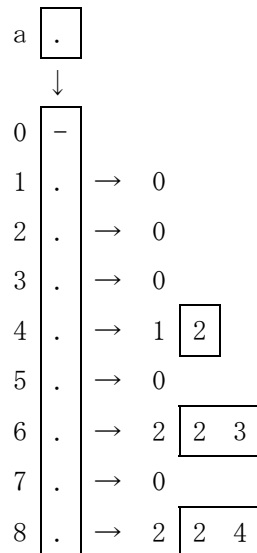
- | | |
|-----------------|---------------|
| 1 --- イタリア | 2 --- イラク |
| 3 --- インド | 4 --- エジプト |
| 5 --- ギリシャ | 6 --- サイパン |
| 7 --- サウジアラビア | 8 --- シンガポール |
| 9 --- スーダン | 10 --- スイス |
| 11 --- ソマリア | 12 --- タイ |
| 13 --- 台湾 | 14 --- ドイツ |
| 15 --- ニューカレドニア | 16 --- ニューヨーク |
| 17 --- ハワイ | 18 --- フィリピン |
| 19 --- ペルー | 20 --- 香港 |
| 21 --- マダガスカル | 22 --- マレーシア |
| 23 --- ロサンゼルス | |

時刻を表示する都市を番号で入力してください。 => 7↵

サウジアラビアの現在の時刻は 5 時 06 分です。

【問題 2】

任意の正整数 m を指定して、 m 以下のすべての正整数 i について、 1 と i 自身を除いた i の正の約数を格納する 2 次元配列 $a[m+1][\]$ のデータ構造を実現しなさい。但し、行数 $m+1$ および各行の長さは可変とし、各 i の約数の前にはそれらが何個あるかを示す数が格納される。すなわち、下図のような `int **a` を構成する。



但し、 $a[0] = \text{NULL}$ とする。 $i > 0$ に対しては、約数の個数 n を $a[i][0]$ に格納し、それら n 個の約数は $a[i][1]$, ..., $a[i][n]$ に格納される。

まず、与えられた m に対して、動的にメモリを割り当てて、このようなデータ構造を作り上げる関数 `void factor(int **a, int m)` をつくりなさい。

`main` 関数では、関数 `factor(a, m)` を呼び出して m 以下の正整数の自明でない約数をすべて $a[m+1][\]$ に格納します。この後でユーザが入力した m 以下の整数 mm に対して、 mm の約数をすべて表示してみましょう。

《ヒント》 値 m が入力された時点で $(m+1)$ 個 (0 から m まで) のポインタへのポインタを格納する `**a` のメモリを動的に確保する。次に、各 $i (\leq m)$ の約数の個数 n を求めて、約数の個数 n とすべての約数を格納するために行 $a[i]$ が要素数 $n+1$ を格納できるようにメモリを動的に確保する。この 2 段階の動的なメモリ確保には、次の関数を利用しなさい。この関数は、 n バイトのメモリ領域を確保してその先頭アドレスを返します。

```
void *getmemory(int n)
{
    void *p = malloc(n);
```

```
    if (p == NULL)
        puts("Memory problem"), exit(1);
    return p;
}
```

特に、この関数の戻り値は総称的なポインタ型 `void *` となっているため、`void *` には任意のポインタがキャスト可能です。例えば、次のように使用します。

```
int *ptr;
...
ptr = (int *) getmemory(100 * sizeof(int));
```

▼出力例

任意の正の整数を入力して下さい (exit = 0) : 100↵

では 100 までの整数の約数を全て求めます ---> 完了!

100 以下の整数を指定して下さい (exit = 0) : 60↵

60 の約数は以下の 10 個です

2 3 4 5 6 10 12 15 20 30

100 以下の整数を指定して下さい (exit = 0) :

【問題3】(三目並べ)

3目並べを行うプログラムを書いてみましょう。プレイヤー(白石 = 1)とコンピュータ(黒石 = 2)が対戦することとし、先攻と後攻はプレイヤーが選択できるようにしなさい。プログラムは次のような手順を実現する必要があります。

- (1) 3×3の競技盤を表示する。空いている目は・, 白石が置かれている目は○, 黒石が置かれている目は●, を表示する。
- (2) 石が置かれるたびに、勝負がついたか(縦, 横, あるいは対角線方向に黒石か白石が3目並んでいる)を調べて、白が勝った, 黒が勝った, 引き分け, まだ勝負がついていない, のいずれかの判定をする。勝負がついていない場合はゲームを続行する。
- (3) プレイヤーは、表示された競技盤を見て、空いているどの目に白石を置くかを縦, 横の目の位置で指定して、キーボードから入力する。空いてない目や無効な目を指定した場合は、やり直しを要求する。
- (4) コンピュータの打つ手は、できるだけ強い打ち方を工夫してください!

▼出力例

3目並べのゲームを始めます。

先攻しますか (y/n) ? n

コンピュータが打ちました。

```

1 2 3
1 . ● .
2 . . .
3 . . .

```

あなたの番です。打つ場所(↓, →)を指定して下さい: 2 2

```

1 2 3
1 . ● .
2 . ○ .
3 . . .

```

コンピュータが打ちました。

```

1 2 3

```

1 . ● .
2 ● ○ .
3 . . .

(途中省略)

あなたの番です。打つ場所(↓, →)を指定して下さい: 2 3

1 2 3
1 ○ ● ●
2 ● ○ ○
3 ○ . ●

コンピュータが打ちました。

1 2 3
1 ○ ● ●
2 ● ○ ○
3 ○ ● ●

引き分けです！

もっと続けますか (y/n) ?

【問題4】（神経衰弱ゲーム）

「神経衰弱ゲーム」のシミュレーションを作成してみましょう。マス目は4×4とします。8種類の動物名が2匹ずつ隠されています。並び順は乱数で発生しましょう。2枚ずつ順番に表にして当たっていれば、表向きのままになります。はずれれば一定時間おいて裏に戻します。最大試行回数を設定して、ゲームを進行します。全部当たれば「やったね！」と祝福し、駄目だった場合は全部表向きにしてから「またね！」とメッセージを表示してみましょう。画面の表示などインタフェースにも工夫してみましょう。

【問題 5】

「ウサギと亀の駆け比べ」のシミュレーションを作成してみましょう。マス目 1 から両者スタートし、ゴールはマス目 70 とします。最初にマス目 70 に到達した方が勝ちで、賞品としてバケツ一杯の新鮮なニンジンとレタスが貰えます。コースは滑りやすく、競技者は時々滑ってしまいます。ウサギと亀は 1 秒毎に、次のルールに従って動きます。

亀	確率 50%	右に 3 マス	(急ぎ足)
	確率 20%	左に 6 マス	(スリップ)
	確率 30%	右に 1 マス	(のろ足)
ウサギ	確率 20%	動かない	(居眠り)
	確率 20%	右に 9 マス	(大ジャンプ)
	確率 10%	左に 12 マス	(大スリップ)
	確率 30%	右に 1 マス	(小ジャンプ)
	確率 20%	左に 2 マス	(小スリップ)

ウサギと亀の位置を表す変数を用意してください。出発点より左にスリップしたときは、マス目 1 に戻します。「スタート」とともに、1 秒毎のウサギと亀の位置を画面上に 70 桁のものさしを用意して、それぞれ H と T の文字で表示してください。また、H と T の位置が重なったときは、亀がウサギに噛み付くので、「痛い！」と表示してください。どちらかがゴールに先に到達するか、同点ゴールの場合は、それぞれ

「やったぜ!!! 亀の勝ち!!!」

「やっぱり、ウサギの勝ち。」

「同着です!!」

と表示して、ゲームを終了してください。

☆もちろん、表示するメッセージは各自工夫してかまいません。

《ヒント》ウサギと亀の動き方は、1~10 までの整数乱数 i を発生して、上記の確率に従って決定してください。例えば、亀の場合、 i が 1~5 のときは「急ぎ足」、 i が 6 か 7 のときは「スリップ」、 i が 8~10 のときは「のろ足」、とすること。