

# 後期中間試験 (1E 情報処理 I)

山本昌志\*

2004 年 12 月 1 日

## 1 制御文

以下の制御文を作成せよ。

[問 1] もし、 $a < b$  ならば、以下を実行する。

- 変数  $a$  に 1 を代入する。
- 変数  $b$  に -1 を代入する。

[問 2] もし、 $a < b$  ならば、

- 変数  $a$  に 1 を代入する。

さもなければ、 $a = b$  ならば、

- 変数  $a$  に 0 を代入する。

さもなければ、

- 変数  $a$  に -1 を代入する。

を実行する制御文を作成せよ。

## 2 配列

C 言語の配列に関する以下の問いに答えよ。

[問 1] 単純型と配列型のデータ構造の違いを簡潔に説明せよ。

[問 2] 単純型に比べて配列型のデータ構造が便利な点を簡潔に説明せよ。

[問 3] 以下の配列を使う場合の宣言を書け。

- 倍精度実数型の配列 `hoge[0] ~ hoge[100]` を使う場合。
- 整数型の配列 `fuga[0][0] ~ fuga[100][100]` を使う場合。

## 3 繰り返し

C 言語の繰り返し文に関する問いである。以下のプログラムを作成せよ。

[問 1] `for` 文を用いて、1 ~ 10000 までの和を計算し、結果を表示する。

[問 2] `while` 文を用いて、1 ~ 10000 までの和を計算し、結果を表示する。

---

\* 国立秋田工業高等専門学校 電気情報工学科

## 4 プログラム作成

### 4.1 温度のデータ処理

以下の動作を行うプログラムがある。

- 11月の毎日の1時間毎の気温の表図1のようなデータファイル (/tmp/1e/temperature.txt) がある。
  - 各行には、その日の1時間毎の24個のデータがある。0時～23時までである。
  - 行数は30行で、11月1日から11月30日を表している。
  - ファイルには、温度のみが書かれている。時刻や日にちは書かれていない。
- 日毎の最高気温と最低気温、平均気温をディスプレイに書き出す。
- 11月の最高気温と最低気温、平均気温をディスプレイに書き出す。

そして、実行結果は、次のようになるとする。このプログラムのソースの [ア] ~ [ソ] に入る適切な文を選択せよ。

```
Temperature November/2003 at Akita
-----
day      max      min      average
=====
 1       20.3     6.0     12.1
 2       20.4     9.5     15.2
 3       22.8    13.9     17.8
 4       14.9     6.4     11.5
 5       18.4     4.3     10.6
 6       17.9    10.5     13.3
  :       :         :         :
  :       :         :         :
25       10.6     7.6     9.2
26        7.0     0.6     4.3
27        7.3    -0.3     2.8
28       12.2     0.1     6.0
29       15.7     7.9    12.2
30       15.1    11.2    13.4
-----
```

```
max(Nov.)    = 22.8
min(Nov.)    = -0.5
average(Nov.) = 9.2
```

表 1: 11月の気温

	0時	1時	2時	3時	...	23時
1日	8.3	7.9	7.5	7.2	...	9.8
2日	9.3	9.2	9.1	9.0	...	6.3
3日	6.2	5.8	5.3	4.9	...	12.0
⋮	⋮	⋮	⋮	⋮	⋮	⋮
30日	4.3	3.9	3.3	2.8	...	3.8

```

#include <stdio.h>

int main(void){
    [ア];
    double temp[31][24];
    double max_day[31], min_day[31], av_day[31];
    double max_nov, min_nov, av_nov;
    double sum_day, sum_nov;
    int dates, hours;
    int i, j;

    /* ----- 日数と時間の設定 -----*/

    dates = 30;
    hours = 24;

    /* ----- ファイルのオープン -----*/

    if([イ]==NULL){
        printf("can not open the file\n");
        return 1;
    }

    /* ----- ファイルからデータを読み込む -----*/

    for(i=1; i<=dates; i++){
        for(j=0; j<=hours-1; j++){
            [ウ];
        }
    }

    /* ----- ファイルのクローズ -----*/

    fclose(fp);

    /* ----- 最大と最小、平均の計算 -----*/

    max_nov = [エ];          /* 11月の仮の最高気温 */
    min_nov = [オ];          /* 11月の仮の最低気温 */
    sum_nov = [カ];          /* 11月の気温の合計の初期値 */

    for([キ]){              /* i は、日にちを表す */

        max_day[i] = [エ];   /* i日の仮の最高気温 */
        min_day[i] = [オ];   /* i日の仮の最低気温 */
        sum_day = [カ];      /* i日の気温の合計の初期値 */

        for([ク]){          /* j は時刻を表す */
            sum_day [ケ];   /* 時刻毎の気温の合計 */
        }
    }
}

```

```

        if([コ]){
            [サ];
        }

        if(temp[i][j] < min_day[i]){
            min_day[i] = temp[i][j];
        }
    }

    [シ];

    sum_nov += av_day[i];

    if(max_day[i] > max_nov){
        max_nov = max_day[i];
    }

    if([ス]){
        [セ];
    }
}

av_nov = [ソ];

/* ----- 結果の表示 ----- */

printf("\n\nTemperature  November/2003 at Akita\n");
printf("-----\n");
printf(" day      max      min      average \n");
printf("=====\n");

for(i=1; i<=dates; i++){
    printf("%3d      %5.11f      %5.11f      %5.11f\n",
        i, max_day[i], min_day[i], av_day[i]);
}

printf("-----\n\n");

printf("max(Nov.)      = %5.11f\n", max_nov);
printf("min(Nov.)      = %5.11f\n", min_nov);
printf("average(Nov.) = %5.11f\n\n", av_nov);

return 0;
}

```

---

選択肢
-----

[ア]	(1) file *fp	(2) FILE *fp_read	(3) FILE fp	(4) FILE *fp
[イ]	(1) fp=fopen("/tmp/1e/temperature.txt","r")	(2) *fp=fopen("/tmp/1e/temperature.txt","r")	(3) fp=fopen("/tmp/1e/temperature.txt","w")	(4) fp=fopen(/tmp/1e/temperature.txt,r)
[ウ]	(1) fscanf(fp, "%lf", temp[i][j])	(2) scanf(fp, "%lf", &temp[i][j])	(3) fscanf(fp, "%lf", &temp[i][j])	(4) fscanf(fp, "%lf", &temp[i,j])
[エ]-[カ]	(1) -9999	(2) 10.0	(3) 0.0	(4) 9999
[キ]	(1) i=0; i<=dates; i++	(2) i=1; i<=dates; i++	(3) j=0; i<=dates; j++	(4) j=1; i<=dates; j++
[ク]	(1) j=1; j<=hours; j++	(2) j=1; j<=hours-1; j++	(3) j=0; j<=hours; j++	(4) j=0; j<=hours-1; j++
[ケ]	(1) += temp[i][j]	(2) = temp[i][j]	(3) -= temp[i][j]	(4) += max_day[j]
[コ]	(1) temp[i][j] < max_day[j]	(2) temp[i][j] > max_day[j]	(3) temp[i][j] < max_day[i]	(4) temp[i][j] > max_day[i]
[サ]	(1) max_day[i] == temp[i][j]	(2) max_day[i] -= temp[i][j]	(3) max_day[i] = temp[i][j]	(4) max_day[i] += temp[i][j]
[シ]	(1) av_day[i] = sum_day*hours	(2) av_day[i] = sum_day/hours	(3) av_day[i] = sum_day+hours	(4) av_day[i] = sum_day-hours
[ス]	(1) min_day[j] < min_nov	(2) min_day[i] < min_nov	(3) min_day[j] > min_nov	(4) min_day[i] > min_nov
[セ]	(1) min_nov == min_day[i]	(2) min_nov -= min_day[i]	(3) min_nov += min_day[i]	(4) min_nov = min_day[i]
[ソ]	(1) sum_nov/dates	(2) sum_nov*dates	(3) sum_nov/hours	(4) sum_nov*hours

---

## 4.2 アクセスカウンター

ホームページを閲覧しているときに、アクセスカウンターと呼ばれる「そのページが何回閲覧されたか」を表示する仕組みがある。これは、ホームページにアクセスしたときにあるプログラムが動くようになっていて、そのプログラムが何回実行されたかを調べている。すなわち、実行回数をファイルに保存しておき、プログラムを実行するときにはそのファイルを読み込んで実行回数に1を加えてまた保存するというを行っているわけだ。

このようなときに利用できるアクセスカウンタープログラムの基本として、プログラムの実行回数をカウントし、「 回目の実行です」と表示するプログラムを作成しなさい。

ただし、

- アクセス数を記述したファイルは、/tmp/1e/up.logにある。
- ファイルをオープンするときは、ファイルがないときのエラー処理を書くこと。

とする。