

〔プログラム〕

```
#include <stdio.h>

typedef struct {
    int Temp ; /* 温度 */
    double Ratio ; /* センサ出力値の比率 (実測値) */
    double Step ; /* 1 当たりの増分 */
} CURVE ;

void SetupCurve( CURVE * , int );
double GetK ( int , CURVE * , int );
#define ITEMS 7

main()
{
    int Degree ;
    double k ;
    CURVE Curve[ ITEMS ] = {
        { -40 , 0.20 , 0.0 } , { -20 , 0.60 , 0.0 } ,
        { -10 , 0.80 , 0.0 } , { 0 , 1.00 , 0.0 } ,
        { 10 , 1.17 , 0.0 } , { 30 , 1.50 , 0.0 } ,
        { 50 , 1.80 , 0.0 } ;
    }

    SetupCurve( Curve , ITEMS );
    printf( " 温度 温度補正係数\n" );
    for( Degree = -40 ; Degree <= 50 ; Degree++ ) {
        k = GetK( Degree , Curve , ITEMS );
        printf( " %3d %4.2f\n" , Degree , k );
    }
}

void SetupCurve( CURVE *p , int Points )
{
    /* 温度補正テーブルを初期化する */
    int i ;
    for( i = 0 ; i < Points - 1 ; i++ ,  ) {
        p->Step = (  ) /
            ( (p+1)->Temp - p->Temp );
    }
}

double GetK( int Temp , CURVE *p , int Points )
{
    /* 温度補正係数 K を返却値として返す */
    int i , j , n ;
    i = 0 ; j = Points - 1 ;
    if ( ( Temp < p->Temp ) || ( Temp >
        (p+j)->Temp ) )
        return 0.0 ; /* 温度範囲外の場合は , 0.0 を返す */
    /* 温度補正テーブルの最後の要素の温度と一致した場合 */
    if ( Temp == ( p+j )->Temp )
        return 1.0 / ( p+j )->Ratio ;
    /* 温度補正テーブルを二分探索法で探索する */
    while( 1 ) {
        n =  ;
        if ( ( Temp >= ( p+n )->Temp ) &&
            ( Temp < ( p+n+1 )->Temp ) ) break ;
        if ( Temp < ( p+n )->Temp ) j = n ;
    }
}

```

```
else  ;
}

p += n ;
return 1.0 / ( p->Ratio + p->Step *
    (  ) );
}

```

設問 プログラム中の に入れる正しい答えを、解答群の中から選べ。

a に関する解答群

- | | |
|-------------|----------|
| ア p-- | イ p++ |
| ウ p->Temp++ | エ p += 3 |
| オ p += 7 | |

b に関する解答群

- | |
|-------------------------------|
| ア (p-1)->Ratio - p->Ratio |
| イ (p-1)->Ratio - (p+1)->Ratio |
| ウ p->Ratio - (p-1)->Ratio |
| エ p->Ratio - (p->Ratio - 1) |
| オ (p+1)->Ratio - p->Ratio |

c に関する解答群

- | | |
|-----------------|-----------------|
| ア (i + j) / 2 | イ (i * j) / 2 |
| ウ (i % j) / 2 | エ (i + j) * 2 |
| オ (i % j) * 2 | |

d に関する解答群

- | | |
|-------------|-------------|
| ア i = 0 | イ i = n - 1 |
| ウ i = n + 1 | エ j = n |
| オ j = n + 1 | |

e に関する解答群

- | | |
|------------------|------------------|
| ア Temp - p->Step | イ Temp - p->Temp |
| ウ Temp + p->Temp | エ p->Temp - n |
| オ p->Temp - Temp | |

平成15年度 秋期 FE 午後問題 C言語

問6 次のcプログラムの説明及びプログラムを読んで、設問に答えよ。

〔プログラムの説明〕

コンサート会場において、連続して確保したい席数を与えられると、座席を決定し、結果を返却値として返すプログラムである。

(1) コンサート会場の座席は、図に示すとおりである。

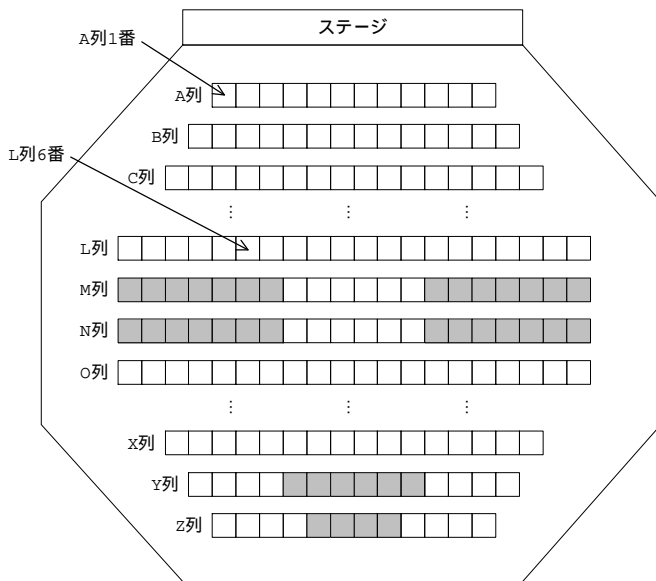


図 コンサート会場の座席配置

座席は、A ~ Z の 26 列からなり、列によって座席数が異なる。各列の座席数は、列 A ~ Z の順に大域の配列 cnum に格納されている。

座席番号は、列と番からなる。番とは、各列についてステージに向かって左から何番目の座席であるかを示す番号のことである。例えば、「L 列 6 番」の座席とは、ステージから 12 列目(L 列)のステージに向かって左から 6 番目の座席を指す。

図の網掛けの席は、予約済である。

(2) 連続して確保したい希望席数を、引数 number に格納してプログラムに渡す。

(3) 座席は、ステージに向かって最前列左端(A 列 1 番)から、同一列に希望席数の入る連続した空きがあるかどうかを調べ、最初に見つかった空きを確保する。A 列で確保できなかった場合は、順次 B 列、C 列、... と後列に対して左端から、希望席数の入る連続した空きがあるかを調べ、最初に見つかった空きを確保する。全席探しても空きが見つからなかった場合には、座席が確保できなかったものとする。

(4) 希望数の座席が確保できた場合は、その座席の列名と、確保できた座席の中で、ステージに向かって最も左端に位置する座席の番号とを格納した座席番号構造体(SEAT)へのポインタを、返却値として返す。確保できなかった場合は、空ポインタ(NULL)を返却値として返す。

(5) 構造体 SEAT の宣言は、次のとおりである。

```
typedef struct {
    char    row;        /* 座席の列名 */
    int     no;        /* 座席の番号 */
} SEAT;
```

(6) コンサート会場の各座席の状況は、大域変数 status に格納されている。座席が確保できた場合には、確保できた座席を予約済にする。

$$status[i][j] = \begin{cases} \text{' '}: \text{座席は空} \\ \text{'R'}: \text{座席は予約済} \end{cases}$$

列 A ~ Z に対応する添字 i の値は 0 ~ 25 になる。
番 1 ~ n に対応する添字 j の値は 0 ~ n - 1 になる。

〔プログラム〕

```
#define MAXNUM 30    /* 列の最大座席数 */
#define ROWNUM 26   /* 列数 */

typedef struct {
    char    row;        /* 座席の列名 */
    int     no;        /* 座席の番号 */
} SEAT;

static char rname[] = {"ABCDEFGHIJKLMN
    OPQRSTUVWXYZ"}, status[ROWNUM][MAXNUM];
static int cnum[ROWNUM] = {12, 14, 16, 18, 20,
    22, 24, 26, 28, 30, 30, 30, 30, 30, 30,
    30, 28, 26, 24, 22, 20, 18, 16, 14, 12};
static SEAT empty;

SEAT *book_seat(int);

SEAT *book_seat(int number)
{
    int ridx, cidx, eidx, flg = 0;

    for (ridx = 0; ridx < ROWNUM; ridx++) {
        for (cidx=0; cidx<=cnum[ridx]-number; cidx++) {
            if (status[ridx][cidx] == ' ') {
                a;
                for (eidx = b;
                    cidx < eidx; eidx--)
                    if (status[ridx][eidx] ==
                        'R') {
                            flg = 0;
                            break;
                        }
                if (flg == 1) break;
            }
        }
    }
}
```

示現塾 プロジェクトマネージャ・テクニカルエンジニア(ネットワーク)など各種セミナーを開催中!!

開催日、受講料、カリキュラム等、詳しくは、<http://zigen.cosmoconsulting.co.jp> 今すぐアクセス!!

```

    }
}
if (flg == 1) break;
}
if (flg == 0)
    return NULL;
for (eidx = cidx + number - 1; cidx <= eidx;
     eidx--)
    status[ridx][eidx] = 'R';
empty.row = ;
empty.no = ;
return &empty;
}

```

設問 プログラム中の に入れる正しい答えを、
解答群の中から選べ。

a に関する解答群

- | | |
|------------------|----------------------|
| ア cidx++ | イ cidx-- |
| ウ cidx += number | エ cidx += number - 1 |
| オ flg = 0 | カ flg = 1 |

b に関する解答群

- | | |
|---------------------|-----------------|
| ア 0 | イ cidx |
| ウ cidx + 1 | エ cidx + number |
| オ cidx + number - 1 | カ number |
| キ number - 1 | |

c に関する解答群

- | | |
|-------------------|-------------------|
| ア ridx | イ ridx + 1 |
| ウ ridx - 1 | エ rname[ridx] |
| オ rname[ridx + 1] | カ rname[ridx - 1] |

d に関する解答群

- | | |
|---------------|---------------|
| ア cidx | イ cidx + 1 |
| ウ cidx - 1 | エ ridx + cidx |
| オ ridx - cidx | |

問 10 次の c プログラムの説明及びプログラムを読んで、
設問 1, 2 に答えよ。

{ プログラムの説明 }

表計算ソフトなどで使用されるデータ交換処理用のプログラムである。

(1) 関数 loadcsv は、(2) に示す様式 (CSV 形式と呼ぶ) のレコードをテキストファイルから読み込み、2次元配列の表に格納する。関数 savecsv は、表内の要素を CSV 形式に戻してテキストファイルに出力する。

(2) このプログラムで扱う CSV形式の仕様は、次のとおりである。

文字列からなるデータをコンマ“,”で区切って並べたものである。

コンマの前後には空白が許される。

文字列は、全体を引用符で囲んでも囲まなくてもよい。ただし、文字列内に空白文字、コンマ“,”を含む場合には、図 1 のように文字列全体を引用符で囲む。文字列内に引用符は含まないものとする。

CSV 形式ファイルには、表の 1 行に対応するデータ群が 1 レコードとして格納されている。

レコードの終端は “ \n ” である。

```
Shop-A, 1, "1,000", "21,000", "3,000"
Shop-B, 2, "4,500", "15,000", "4,000"
```

図 1 CSV 形式のデータ例 (2 レコード)

(3) レコード内の文字列は、動的に確保した領域に格納し、表の要素にはその領域へのポインタを格納する。

(4) 引用符で囲まれたデータ形式の場合は、動的に確保した領域に前後の引用符を除いた文字列を格納する。

(5) 対応するデータがない要素には、NULL を格納する。

(6) 図 1 のレコードを格納した表を図 2 に示す。

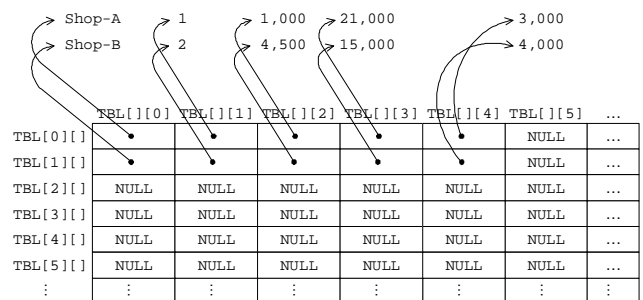


図 2 CSV 形式のデータを格納した表の例 (2 レコード)

(7) ここで使用される文字の種類は、JIS X 0201 (7ビット及び 8ビットの情報交換用符号化文字集合)の文字とする。

(8) 表の行数及び列数の最大値は、ともに 64 とし、レコードの文字長は 255 以下とする。

(9) csv 形式ファイル内のレコードに記述の誤りはないものとし、ファイル入出力でのエラーも発生しないものとする。

(10) 各関数の仕様は、次のとおりである。

関数 loadcsv

形式: int loadcsv(char fname[], char *tbl[][TBLSZ]);

引数: fname 入力ファイル名へのポインタ
tbl 表(2次元配列)へのポインタ(ここで、TBLSZ は、表の行数及び列数の最大値)

返却値: 表に格納したレコード数

関数 savecsv

形式: void savecsv(char fname[], char *tbl[][TBLSZ], int ln);

引数: fname 出力ファイル名へのポインタ
tbl 表(2次元配列)へのポインタ
ln 出力するレコード数

【プログラム】

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define TBLSZ 64
#define RECSZ 256

int loadcsv( char*, char*[][TBLSZ] );
void savecsv( char*, char*[][TBLSZ], int );

int loadcsv( char fname[], char *tbl[][TBLSZ] )
{
    FILE *fp;
    char buf[RECSZ], *cp, *dp;
    int col, row, len;

    for ( row = 0; row < TBLSZ; row++ ) {
        for ( col = 0; col < TBLSZ; col++ )
            tbl[row][col] = NULL;
    }
    fp = fopen( fname, "r" );
    for ( row = 0; fgets( buf, RECSZ, fp ) !=
        NULL; row++ ) {
        col = 0;
        for ( cp = buf; *cp != '\n'; cp++ ) {
            while ( *cp == ' ' ) cp++;
            /* 空白の読み飛ばし */

            if ( *cp == '\n' ) break;
            else if ( *cp == ',' )
```

```
                a ; /* コンマの処理 */
            else {
                if ( *cp == '"' ) { /* データが
                    引用符で囲まれている場合 */
                    dp = b;
                    for ( len = 0; *cp != '"';
                        len++, c );
                }
                else {
                    dp = d;
                    for ( len = 0; *cp != '\n';
                        len++, c )
                        if ( ( *cp == ' ' ) || ( *cp
                            == ',' )
                            || ( *cp == '\n' ) ) break;
                    e;
                }
            }
            if ( len != 0 ) { /* データが空で
                ない場合 */
                tbl[row][col] = malloc
                    ( len+1 );
                /* 動的に領域を確保 */
                strncpy( tbl[row][col],
                    dp, len );
                *(tbl[row][col]+len) = '\0';
            }
        }
    }
    fclose( fp );
    return row;
}

void savecsv( char fname[], char *tbl[][TBLSZ],
    int ln )
{
    FILE *fp;
    int row, col, last;

    fp = fopen( fname, "w" );
    for ( row = 0; row < ln; row++ ) {
        for ( last = TBLSZ-1; last >= 0; last-- )
            if ( tbl[row][last] != NULL ) break;
        for ( col = 0; col <= last; col++ ) {
            if ( tbl[row][col] != NULL ) {
                fputs( "\"", fp );
                fputs( tbl[row][col], fp );
                fputs( "\"", fp );
            }
            if ( col < last ) fputs( ",", fp );
        }
        fputs( "\n", fp );
    }
    fclose( fp );
}
```

設問1 プログラム中の [] に入れる正しい答えを、解答群の中から選べ。なお、解答は重複して選んでもよい。

a に関する解答群

- ア col++ イ dp = NULL ウ dp++
エ len++ オ tbl[row][col] = NULL

b ~ e に関する解答群

- ア cp イ cp--
ウ cp+1 エ ++cp

設問2 次の CSV 形式の 2 レコードを関数 loadcsv に
よって表に格納し、それから関数 savecsv によっ
て CSV 形式で出力する。出力結果の説明として適切
なものを、解答群の中から選べ。

"Taro Yamada", 43, "175.5", M, "550,000"
Hanako-Yamada, 12, "156.7", F,

解答群

ア すべてのデータに対して、引用符で囲んだものをコ
ンマで区切って出力する。各レコードの末尾には、不
要なコンマは出力されない。

"Taro Yamada", "43", "175.5", "M", "550,000"
"Hanako-Yamada", "12", "156.7", "F"

イ すべてのデータに対して、引用符で囲んだものをコ
ンマで区切って出力する。出力レコードのデータ個数
は最大のものにそろえられる。

"Taro Yamada", "43", "175.5", "M", "550,000"
"Hanako-Yamada", "12", "156.7", "F", ""

ウ 入力時と同じ形式で出力する。出力レコードのデー
タ個数は最大のものにそろえられる。

"Taro Yamada", 43, "175.5", M, "550,000"
Hanako-Yamada, 12, "156.7", F,

エ 入力時と同じ形式で出力する。レコードの末尾には、
不要なコンマは出力されない。

"Taro Yamada", 43, "175.5", M, "550,000"
Hanako-Yamada, 12, "156.7", F

平成15年度 春期 FE 午後解答 C言語

問6

設問1

a - ク b - オ c - エ

設問2

ア

問10

設問

a - イ b - オ c - ア d - ウ e - イ

平成15年度 秋期 FE 午後解答 C言語

問6

設問

a - カ b - オ c - エ d - イ

問10

設問1

a - ア b - エ c - エ d - ア e - イ

設問2

ア