

平成 16 年度 春期 F E 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〕

図に示すように、2次元平面の中に n 個 ($3 \leq n \leq 20$) の点を与えられたときに、これらの点すべてを含む円のうち、半径が最小である円の中心座標と半径を求めるプログラムである。

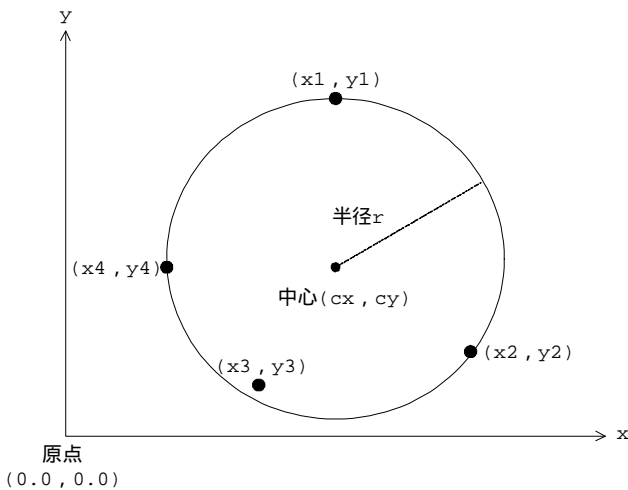


図 四つの点 $(x1, y1) \sim (x4, y4)$ を含む円

(1) 各点の位置は、 x 座標と y 座標で与えられる。各座標値はともに 10^4 を超えない正の値である。

(2) ここでは、次のアルゴリズムによって、求める円の中心を探し出している。

仮中心点を原点 $(0.0, 0.0)$ に、移動係数を 0.5 に設定する。

n 個の点の中で、仮中心点から最も遠い位置にある点（最遠点）の方向に向かって、一定の距離（以下、移動距離と呼ぶ）だけ仮中心点を移動する。このときの移動距離は、移動前の仮中心点から最遠点までの距離に移動係数を乗じた値とする。これを一定回数（ここでは 100 回）繰り返した後にへ移る。

なお、繰り返しの途中で、仮中心点の移動距離が一定の値 (10^{-9}) 以下になった場合には、移動前の仮中心点を円の中心座標とみなし、処理を終了する。

移動係数を $\frac{1}{2}$ 倍にして、へ戻る。

(3) 関数 solvcircle の仕様は、次のとおりである。

```
void solvcircle( double pt_x[],
                double pt_y[], double *cx,
                double *cy, double *r );
```

・機能：引数で受け取った点の座標値を基に、これらの点を内包する最小円の中心座標 (x 座標と y 座標) と半径を求める。

・引数：pt_x 対象となる各点の x 座標を格納した配列変数
(終端には -1.0 が格納されている。)

pt_y 対象となる各点の y 座標を格納した配列変数
(終端には -1.0 が格納されている。)

cx 求めた円の中心点の x 座標を格納する変数へのポインタ

cy 求めた円の中心点の y 座標を格納する変数へのポインタ

r 求めた円の半径を格納する変数へのポインタ

(4) 関数 distance の仕様は、次のとおりである。

```
double distance( double x1, double y1,
                 double x2, double y2 );
```

・機能：引数で指定された 2 点間の距離を求める。

・引数：x1, y1 一方の点の x 座標 ($x1$) と y 座標 ($y1$)
x2, y2 他方の点の x 座標 ($x2$) と y 座標 ($y2$)

・返却値：2 点 $(x1, y1)$ と $(x2, y2)$ の距離

(5) 関数 main は、4 個の点座標が与えられたときの関数 solvcircle の使用例を示している。

(6) プログラムでは、解を 10^{-8} までの精度で求める。

〔プログラム〕

```
#include <stdio.h>
#include <math.h>

#define CONVRGV 1.0e-9
#define EOD -1.0
#define LOOPMAX 100
#define ON 1
#define OFF 0

void solvcircle( double[], double[], double*,
                double*, double* );
double distance( double, double, double,
                double );

main()
{
    double pos_x[] = { 10.0, 50.0, 50.0,
                      100.0, EOD };
    double pos_y[] = { 50.0, 10.0, 100.0,
                      50.0, EOD };
```

```
double x, y, r;

solvcircle( pos_x, pos_y, &x, &y, &r );
printf( " 円の中心座標 ( %12.8f , %12.8f )
        半径 = %12.8f\n", x, y, r );
}

void solvcircle( double pt_x[], double pt_y[],
                double *cx, double *cy, double *r )
{
    double mvrate=0.5, maxdist, dist;
    double xsv=0.0, ysv=0.0;
    int k, i, t, cvgflg;

    *cx = *cy = 0.0; /* 仮中心点の初期座標 */
    cvgflg = OFF;
    while ( cvgflg == OFF ) {
        for ( t=0; t < LOOPMAX; t++ ) {
            maxdist = 0.0;
            for ( i=0; pt_x[i] != EOD; i++ ) {
                dist = distance( *cx, *cy,
                                pt_x[i], pt_y[i] );
                if ( dist > maxdist ) {
                    maxdist = dist;
                    k = i;
                }
            }
            *cx += ( pt_x[k] -  )
                  * mvrate;
            *cy += ( pt_y[k] -  )
                  * mvrate;
            if (  <= CONVRGV ) {
                *cx = xsv;
                *cy = ysv;
                cvgflg = ON;
                break;
            }
            xsv = *cx;
            ysv = *cy;
        }
        mvrate /= 2.0;
    }
    *r =  ;
}

double distance( double x1, double y1,
                double x2, double y2 )
{
    return sqrt( (x2-x1)*(x2-x1) +
                (y2-y1)*(y2-y1) );
}

```

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, b, d に関する解答群

- ア *cx
- イ *cy

- ウ dist
- エ distance(pt_x[0], pt_y[0], *cx, *cy)
- オ distance(pt_x[0], pt_y[0], xsv, ysv)
- カ maxdist
- キ pt_x[0]
- ク pt_y[0]

c に関する解答群

- ア distance (pt_x[k], pt_y[k], *cx, *cy)
- イ distance (pt_x[k], pt_y[k], xsv, ysv)
- ウ distance (xsv, ysv, *cx, *cy)
- エ maxdist
- オ maxdist - dist

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 1 の説明】

c 言語の仕様で記述された空でないソースプログラムを
標準入力から読み込んで、注釈を除去して標準出力へ出力
するプログラムである。

(1) ソースプログラムの表記に関する説明

このプログラムで扱う注釈は、文字定数、文字列リ
テラル又は注釈の中を除いて、“/*”で始まり“*/
”で終わる文字の列である。

使用可能文字の種類は、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)である。
ただし、“¥”は“\”と表記する。

次のような記述はない。

() 注釈の入れ子記述

例 /* aaaaa /* bbbbbb */ ccccc */

() 図形文字の3文字表記

??= ??(??/ ??' ??< ??> ??) ??! ??-

文法上の誤りはない。

(2) プログラム 1 では、次の手順で注釈除去を行ってい
る。ただし、文字定数、文字列リテラル及び注釈の解析
を単純に処理しているので、ソースプログラムの記述に
よっては、これらを正しく認識できずに誤った動作を起
こす場合がある。

一重引用符又は二重引用符を検出すると、文字定数
又は文字列リテラルの開始と解釈し、関数 quote を

使用して、対応する一重引用符又は二重引用符を検出するまで、文字の列を読み込んでそのまま出力する。

“/*”を検出すると、注釈の開始と解釈し、最初に現れる“*/”までの文字の列を読み飛ばす。

(3) プログラム 1 による注釈除去の実行例を図に示す。

入力ソースプログラム

```
/* This program uses fgets to display
 * a line from a file on the screen. */
#include <stdio.h>
int main( void )
{
    FILE *stream; /* file pointer */
    char line[100]; /* input stream */

    if( (stream = fopen( "crt_fgets.txt",
                        "r" )) != NULL )
    {
        if( fgets( line, 100, stream ) == NULL)
            printf( "fgets error\n" ); /* error
                                     message */
        else
            printf( "%s", line);
        fclose( stream );
    }
}
```

注釈を除去した後の出力結果

```
#include <stdio.h>
int main( void )
{
    FILE *stream;
    char line[100];

    if( (stream = fopen( "crt_fgets.txt",
                        "r" )) != NULL )
    {
        if( fgets( line, 100, stream ) == NULL)
            printf( "fgets error\n" );
        else
            printf( "%s", line);
        fclose( stream );
    }
}
```

図 注釈除去の実行例

{プログラム 1}

```
#include <stdio.h>
void quote( char );

main()
{
    int c1, c2;
```

```
while ( (c1 = getchar()) != EOF ) {
    /* 一重引用符の検出 */
    if ( c1 == '\'' ) quote( '\'' );
    /* 二重引用符の検出 */
    else if ( c1 == '\"' ) quote( '\"' );
    /* 斜線の検出 */
    else if ( c1 == '/' ) {
        c2 = getchar();
        /* 次の文字がアスタリスクのとき */
        if ( c2 == '*' ) {
            /* 注釈文字列の除去 */
            while ( 1 ) {
                while ( (c1 = getchar()) !=
                        '*' );
                c2 = getchar();
                if ( c2 == '/' ) break;
            }
        }
        /* その他の場合 */
        else {
            putchar(c1);
            putchar(c2);
        }
    }
    else putchar(c1); /* 読み込んだ1文字を
                       そのまま出力*/
}
}
```

```
void quote( char c )
{ /* 文字定数及び文字列リテラルの抽出 */
  char cc;

  putchar(c);
  while ( (cc = getchar()) != c ) putchar(cc);
  putchar(cc);
}
```

設問1 プログラム 1 に入力すると誤った動作を起こす記述を、解答群の中から選べ。

解答群

- ア /* "aaaaaaa" */
- イ /* aaa 'a' */
- ウ if (c == '\'') {
- エ printf(" \' ");
- オ printf("aaa /* comment */ \n");

{プログラム 2 の説明}

プログラム 2 は、プログラム 1 の説明 (2) で指摘した問題点を解決するために、次のように書き換えたものである。

- (1) 文字定数，文字列リテラル，注釈の三つのモードに分けて処理する。
- (2) “文字定数モード”は，一重引用符の出現によって，モードのオンとオフが切り替わる。ただし，“\”による拡張表記として記述されたもの，文字列リテラル内及び注釈内に記述されたものは該当しない。
- (3) “文字列リテラルモード”は，二重引用符の出現によって，モードのオンとオフが切り替わる。ただし，“\”による拡張表記として記述されたもの，文字定数内及び注釈内に記述されたものは該当しない。
- (4) “注釈モード”は，“/*”及び“*/”の出現によって，モードのオンとオフが切り替わる。ただし，文字定数内及び文字列リテラル内に記述されたものは該当しない。

〔プログラム 2〕

```
#include <stdio.h>
main()
{
    int c1, c2;
    int c_mode = 0;
        /* 注釈モードをオフに初期化 */
    int quote1 = 0;
        /* 文字定数モードをオフに初期化 */
    int quote2 = 0;
        /* 文字列リテラルモードをオフに初期化 */

    for ( c1 = getchar(); ( c2 = getchar() ) !=
        EOF; c1 = c2 ) {
        if ( !c_mode ) { /* 注釈モードがオフのとき */
            /* 文字定数又は文字列リテラルの中で文字 \ を
                検出 */
            if (  && c1 == '\\') {
                putchar(c1);
                putchar(c2);
                c2 = getchar();
                continue;
            }
            /* 文字列リテラル以外のところで一重引用符を
                検出 */
            else if ( !quote2 && c1 == '\'' )
                 ;
            /* 文字定数以外のところで二重引用符を検出 */
            else if ( !quote1 && c1 == '\"')
                 ;
            /* 文字定数及び文字列リテラル以外で /と * を
                検出 */
            else if (  && c1 == '/' &&
                c2 == '*' ) {
                 ;
                c2 = getchar();
                continue;
            }
            putchar(c1);
        }
    }

    else {
```

```
        if ( c1 == '*' && c2 == '/' ) {
            /* 注釈の終端か? */
             ;
            c2 = getchar();
        }
    }
    putchar(c1);
}
```

設問2 プログラム 2 中の に入れる正しい答えを，解答群の中から選べ。

a, d に関する解答群

- ア !quote1
- イ !quote2
- ウ (!quote1 || !quote2)
- エ (!quote1 && !quote2)
- オ (quote1 || quote2)
- カ (quote1 && quote2)

b, c, e に関する解答群

- ア c_mode = !c_mode
- イ c_mode = quote1 && quote2
- ウ quote1 = !quote1
- エ quote1 = !quote2
- オ quote1 = quote2
- カ quote2 = !quote1
- キ quote2 = !quote2
- ク quote2 = quote1

平成16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〕

CGI などへのリクエストの際に文字列パラメータを URL に含める場合には，その文字列パラメータを決められた規則に従って変換し送出する必要がある。このプログラムは，それに用いる変換プログラム URLEncode である。

- (1) 変換対象の文字列パラメータは，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) の文字で構成される。また，文字列パラメータの途中にナル文字 (文字符号 0x00) は含まれないものとする。

(2) 変換規則は次のとおりである。

英数文字(0x30 ~ 0x39, 0x41 ~ 0x5A, 0x61 ~ 0x7A)及び“@”(0x40), “*” (0x2A), “-”(0x2D), “.”(0x2E), “_”(0x5F)は変換しない(これらの文字を以降、無変換文字と呼ぶ)

に含まれない文字は, “%”の後に文字符号の2けたの16進表記を続けた3文字に変換する。例えば, 文字符号 0x5E の文字は “%5E” と変換する。

(3) 関数 URLEncode の仕様は, 次のとおりである。

形式: void URLEncode(unsigned char *input, unsigned char *output)

引数: input 変換前の文字列が格納されている文字型配列へのポインタ

output 変換後の文字列を格納する文字型配列へのポインタ

(4) output が指す配列は, 変換後の文字列を格納するのに十分な領域が確保されているものとする。

(5) 例えば, “Hi!”という文字列パラメータを変換した結果は図のとおりとなる。

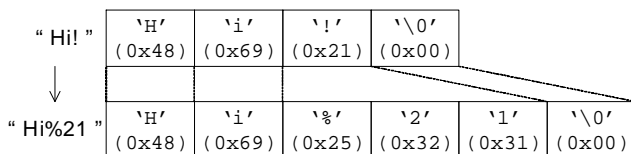


図 文字列パラメータの変換例

(6) 次の関数があらかじめ用意されている。

形式: int replaceChar(unsigned char c)

引数: c

返却値: 文字 c が無変換文字の場合は 0 を, それ以外の場合は 1 を返す。

〔プログラム〕

```
int replaceChar( unsigned char );
void URLEncode( unsigned char *input,
                unsigned char *output ) {
    const unsigned char chars[] =
        "0123456789ABCDEF";
    while( *input != '\0' ) {
        if ( replaceChar(  ) ) {
            *output++ = '%';
            *output++ = chars[  ];
        }
    }
}
```

```
*output++ = chars[  ];
} else {
    *output++ = *input;
}
;
}
*output = '\0';
}
```

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, d に関する解答群

- | | |
|--------------|--------------|
| ア &input | イ (&input)++ |
| ウ (*input)++ | エ **input |
| オ **input++ | カ *input |
| キ input | ク input++ |

b, c に関する解答群

- | | |
|-----------------|-----------------|
| ア *input << 4 | イ *input >> 4 |
| ウ *input & 0x0F | エ *input & 0xF0 |
| オ *input ^ 0x0F | カ *input ^ 0xF0 |
| キ *input 0x0F | ク *input 0xF0 |

問10 次の C プログラムの説明及びプログラムを読んで, 設問に答えよ。

〔プログラムの説明〕

画面上に表示されるマーカを操作することによって, 線画を描く関数 execute である。

(1) 水平方向に 800 画素, 垂直方向に 600 画素からなるビットマップ画面がある。画面の座標系を図 1 に示す。画面上には, 位置と進行方向の情報をもつマーカ(図 1 中の・)が表示されている。マーカの進行方向は, 上, 下, 左, 右の 4 通りのいずれかである。マーカが移動すると, その軌跡が描画される。

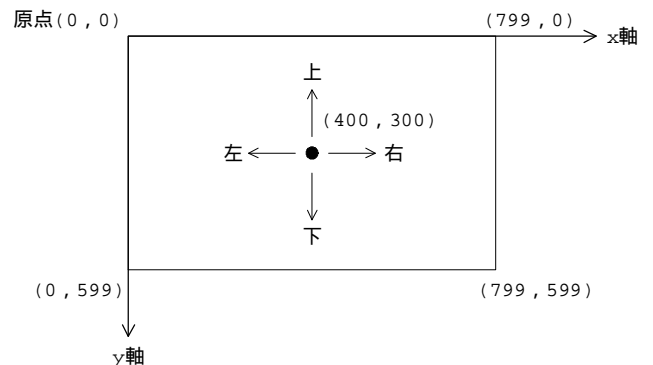


図 1 画面の座標系とマーカの初期状態

- (2) マーカは MARKER 型の構造体 mark で表現する。プログラムの実行開始時には、マーカの位置は (400, 300), 進行方向は上に設定される。

```
typedef struct {
    int x; /* マーカの x 座標 */
    int y; /* マーカの y 座標 */
    int dir; /* マーカの進行方向
            0:右 1:上 2:左 3:下 */
} MARKER;
MARKER mark = {400, /* マーカの初期の x 座標 */
               300, /* マーカの初期の y 座標 */
               1 /* マーカの初期の
                 進行方向(上)*/};
```

- (3) マーカを操作するための命令が定義されている。命令は、命令コードと値から構成され、構造体 INST で表現する。

```
typedef struct {
    char code; /* 命令コード */
    int val; /* 値 */
} INST;
```

命令は、構造体 INST の配列である insts の先頭から実行する順番に格納しておく。

- (4) 命令コードとその説明を次に示す。

命令コード	説明
{	次の命令から対となる命令コード '}' の直前の命令までを val 回繰り返す。val は 1 以上の整数値である。
t	マーカの進行方向を反時計回りに 90 ° x val だけ変更する。val は非負の整数値である。
f	現在の進行方向にマーカを val 画素分だけ進め、移動元から移動先まで線分を描く。val は任意の整数値である。
}	繰り返す一連の命令の終了を示す。val は参照しない。
\0	命令列の終了を示す。val は参照しない。

- (5) 例えば、図 2 のように構造体配列 insts に命令を格納して、関数 execute を実行した場合の出力は図 3 のとおりである。ただし、図 3 中の座標値は説明のために添えたものであり、実際は出力されない。

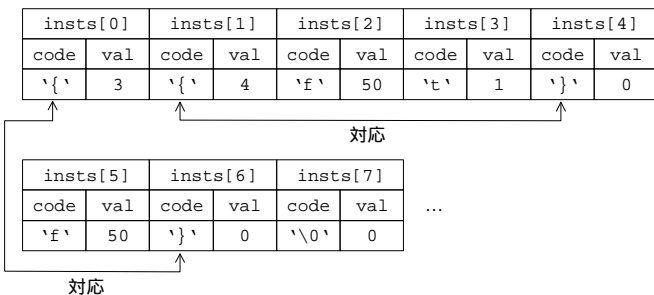


図 2 構造体配列 insts に格納されている命令の例

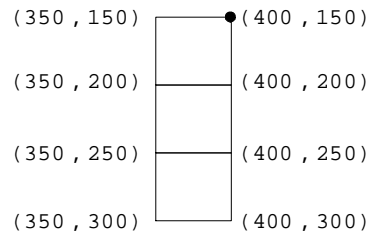


図 3 図 2 の例の実行結果

- (6) 線分を描くために次の関数が用意されている。

```
void drawLine( int x1, int y1,
              int x2, int y2 );
```

機能: 座標 (x1, y1) と座標 (x2, y2) を結ぶ線分のうち、画面領域に含まれる部分を描画する。

- (7) マーカに関する次の関数が用意されている。

```
void eraseMarker( MARKER mark );
```

機能: マーカが画面領域内にあるとき、マーカの表示を消去する。

```
void paintMarker( MARKER mark );
```

機能: マーカが画面領域内にあるとき、マーカを描画する。

- (8) マーカが画面領域外に移動し画面上に表示されなくても、その位置座標及び進行方向の情報は保持される。

【プログラム】

```
#define INSTSIZE 100 /* 命令数の上限 */
#define STACKSIZE 50 /* {}の入れ子の上限 */

typedef struct {
    int x; /* マーカの x 座標 */
    int y; /* マーカの y 座標 */
    int dir; /* マーカの進行方向 0:右 1:上 2:左 3:下 */
} MARKER;

typedef struct {
    char code; /* 命令コード */
    int val; /* 値 */
} INST;

typedef struct {
    int opno; /* 繰返しの開始が定義されている配列
              insts の要素番号 */
    int rest; /* 繰返しの残り回数 */
} STACK;

void drawLine( int, int, int, int );
```

```

void eraseMarker( MARKER );
void paintMarker( MARKER );

INST insts[INSTSIZE]; /* 命令を格納するための構造体配列
                        */
MARKER mark = {400, /* マーカの初期の x 座標 */
               300, /* マーカの初期の y 座標 */
               1 /* マーカの初期の進行方向(上) */
};

void execute(){

    STACK stack[STACKSIZE];
    int opno = 0; /* 実行する命令の配列 insts の
                  要素番号 */
    int spt = -1; /* スタックポインタ */
    int dx, dy;

    paintMarker( mark );

    while( insts[opno].code != '\0' ){
        switch( insts[opno].code ){
            case 'i':
                stack[  ].opno = opno;
                stack[spt].rest = insts[opno].val;
                break;
            case 't':
                mark.dir = ;
                break;
            case 'f':
                eraseMarker( mark );
                dx = ( mark.dir % 2 == 0 ) ?
                    ;
                dy = ( mark.dir % 2 == 0 ) ?
                    ;
                drawLine( mark.x, mark.y,
                        mark.x + dx, mark.y + dy );
                mark.x += dx;
                mark.y += dy;
                paintMarker( mark );
                break;
            case '':
                if ( stack[spt].rest  ){
                    opno = stack[spt].opno;
                    stack[spt].rest--;
                } else {
                    ;
                }
                break;
        }
        ;
    }
}

```

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 ++spt イ --spt ウ spt
エ spt++ オ spt--

b に関する解答群

ア (mark.dir + insts[opno].val) % 2
イ (mark.dir + insts[opno].val) % 3
ウ (mark.dir + insts[opno].val) % 4
エ mark.dir + insts[opno].val
オ mark.dir + insts[opno].val % 2
カ mark.dir + insts[opno].val % 3
キ mark.dir + insts[opno].val % 4

c, d に関する解答群

ア (1 - mark.dir) * insts[opno].val : 0
イ (2 - mark.dir) * insts[opno].val : 0
ウ (mark.dir - 1) * insts[opno].val : 0
エ (mark.dir - 2) * insts[opno].val : 0
オ 0 : (1 - mark.dir) * insts[opno].val
カ 0 : (2 - mark.dir) * insts[opno].val
キ 0 : (mark.dir - 1) * insts[opno].val
ク 0 : (mark.dir - 2) * insts[opno].val
ケ 0 : mark.dir * insts[opno].val
コ mark.dir * insts[opno].val : 0

e に関する解答群

ア < 0 イ < 1 ウ == 0
エ > 0 オ > 1

f, g に関する解答群

ア mark.dir++ イ mark.dir--
ウ opno++ エ opno-
オ spt++ カ spt--

平成16年度 春期 FE 午後解答 C言語

問6

設問

a - ア b - イ c - ウ d - カ

問10

設問1

ウ

設問2

a - オ b - ウ c - キ
d - エ e - ア

平成16年度 秋期 FE 午後解答 C言語

問6

設問

a - カ b - イ c - ウ d - ク

問10

設問

a - ア b - ウ c - ア d - ク
e - オ f - カ g - ウ