

平成14年度 春期 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〕

水平方向に512画素、垂直方向に512画素からなるビットマップ画面がある。水平方向をx座標、垂直方向をy座標とする。座標の割当てを図に示す。

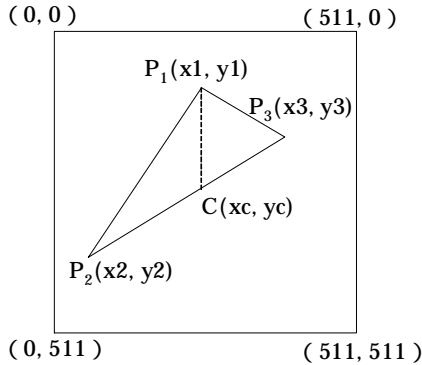


図 ビットマップ画面の座標の割当て

このビットマップ画面に三角形を描画する。三角形は、3頂点 $P_1(x_1, y_1)$ 、 $P_2(x_2, y_2)$ 、 $P_3(x_3, y_3)$ によって与えられる。三角形の各辺、及び辺で囲まれた閉領域内の画素をすべて点灯する再帰的な関数 DrawTrg の処理手順を、次に示す。

- (1) 与えられた3頂点が隣接しているかどうかを検査する。
- (2) 隣接しているときは、3頂点の画素を点灯して処理を終了する。
- (3) 隣接していないときは、三平方の定理を用いて、各辺の長さの2乗値を求める。
- (4) 最も長い辺を選び、この辺の中点Cの座標(xc, yc)を求める。この中点によって二つの三角形に分割する。例えば、図のように辺 P_2P_3 が最長である場合には、三角形 P_1CP_2 と三角形 P_1CP_3 に分割する。
- (5) 分割したそれぞれの三角形に、再帰的にこの関数 DrawTrg を適用する。

画素を点灯するために、次の関数が用意されている。

```
void gSetPixel( int sx, int sy );
```

機能：座標 (sx, sy) の画素を点灯する。

〔プログラム〕

```
void gSetPixel( int, int );
double GetMax( double, double, double );
double GetMin( double, double, double );

double GetMax( double a, double b, double c )
```

```
{
    if ( ( a >= b ) && ( a >= c ) ) return a;
    if ( b >= c ) return b;
    return c;
}

double GetMin( double a, double b, double c )
{
    if ( ( a <= b ) && ( a <= c ) ) return a;
    if ( b <= c ) return b;
    return c;
}

void DrawTrg( double x1, double y1, double x2,
              double y2, double x3, double y3 )
{
    double Len1, Len2, Len3;
    double dx, dy;

    dx = GetMax( x1, x2, x3 ) - GetMin( x1, x2, x3 );
    dy =  - GetMin( y1, y2, y3 );
    if ( ( dx <= 1.0 ) && ( dy <= 1.0 ) ) {
        gSetPixel( (int)x1, (int)y1 );
        gSetPixel( (int)x2, (int)y2 );
        gSetPixel( (int)x3, (int)y3 );
        return;
    }
    Len1 = ( x2-x1 )*( x2-x1 ) + ( y2-y1 )*( y2-y1 );
    Len2 = ( x3-x2 )*( x3-x2 ) + ( y3-y2 )*( y3-y2 );
    Len3 = ( x1-x3 )*( x1-x3 ) + ( y1-y3 )*( y1-y3 );

    if (  ) {
        DrawTrg( x1, y1, (x1+x2)/2, (y1+y2)/2,
                x3, y3 );
        DrawTrg( x3, y3, (x1+x2)/2, (y1+y2)/2,
                x2, y2 );
    }
    else if ( Len2 >= Len3 ) {
        
    }
    else {
        DrawTrg( x1, y1, (x1+x3)/2, (y1+y3)/2, x2,
                y2 );
        DrawTrg( x2, y2, (x1+x3)/2, (y1+y3)/2, x3,
                y3 );
    }
}
}
```

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 GetMax(x1, x2, x3)
- イ GetMin(x1, x2, x3)
- ウ GetMax(x1, y1, y2)
- エ GetMax(y1, x1, x2)

オ GetMax(y1, y2, y3)

カ GetMin(y1, y2, y3)

b に関する解答群

ア Len1 <= Len2

イ Len1 <= Len3

ウ Len2 <= Len3

エ (Len1 >= Len2) && (Len1 >= Len3)

オ (Len1 <= Len2) && (Len1 <= Len3)

c に関する解答群

ア DrawTrg(x1, y1, (x1+x3)/2, (y1+y3)/2, x3, y3);
DrawTrg(x2, y2, (x1+x3)/2, (y1+y3)/2, x1, y1);

イ DrawTrg(x1, y1, (x2+x3)/2, (y2+y3)/2, x3, y3);
DrawTrg(x2, y2, (x2+x3)/2, (y2+y3)/2, x1, y1);

ウ DrawTrg(x2, y2, (x1+x3)/2, (y1+y3)/2, x3, y3);
DrawTrg(x3, y3, (x1+x3)/2, (y1+y3)/2, x1, y1);

エ DrawTrg(x2, y2, (x2+x3)/2, (y2+y3)/2, x3, y3);
DrawTrg(x3, y3, (x2+x3)/2, (y2+y3)/2, x1, y1);

オ DrawTrg(x3, y3, (x1+x3)/2, (y1+y3)/2, x1, y1);
DrawTrg(x3, y3, (x1+x3)/2, (y1+y3)/2, x2, y2);

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〕

マクロ機能をもつあるプログラム言語で書かれたプログラムが、元ファイル source_file に格納されている。この source_file 中のマクロ名を展開し、展開ファイル dest_file を生成する。

(1) 元ファイルには、マクロ定義とマクロ名を使用するプログラムのコードが格納されている。

(2) 展開ファイルには、マクロ名を展開したプログラムのコードを出力する。ただし、元ファイルのマクロ定義は、出力しない。

(3) マクロ定義は、次の形式とする。

\$\$STRDEF マクロ名 展開コード

\$\$STRDEF は、必ず 1 けた目から始まり、\$\$STRDEF とマクロ名の間、及びマクロ名と展開コードの間は、一つの空白文字で区切られる。

(4) 一つのマクロ名に対するマクロ定義は、1 回だけ行うことができる。

(5) 元ファイル中で、マクロ定義以降のコードに対して、区切り文字で区切られたマクロ名の部分を展開コードに置換する。ただし、マクロ定義の行は、置換の対象とはしない。

(6) 区切り文字は、次の ~ であり、文字型の配列 delmchar に格納されている。

空白文字

図形文字 !"#\$%&'()*+,-./:;<=>?[\]^_{`|}~

改行文字

char delmchar[] =

"!\"#\$%&'()*+,-./:;<=>?[\]^_{`|}~\n";

注 delmchar の先頭の要素には、空白文字が格納されている。

(7) マクロ名及び展開コードは、区切り文字を含まない。

(8) 元ファイル中に現れるマクロ定義の個数は、20 までとする。

(9) マクロ名及び展開コードは、31 文字までとする。

(10) 展開の前も後も、各行は改行文字で終わり、文字数は 255 文字までとする。

(11) プログラム中で使用している関数 strchr の仕様は、次のとおりである。

char *strchr(const char *s, int c)

機能： s で指定される文字列の中の c (char 型に変換する。) の最初の出現を捜す。文字列の終端を示すナル文字は、文字列の一部とみなす。

返却値： 捜し出した文字へのポインタを返す。文字列の中に、その文字が現れない場合、空ポインタ (NULL) を返す。

(12) このプログラムの実行例は、次のとおりである。

| 元ファイル | 展開ファイル |
|----------------------------|-------------------|
| \$\$STRDEF \$\$TITLE HELLO | |
| \$\$STRDEF \$\$SIGN BILLY | NSG START |
| \$\$STRDEF \$\$LENG 8 | OUT HELLO,6 |
| | IN BUF,8 |
| MSG START | OUT BUF,8 |
| OUT \$\$TITLE,6 | OUT BILLY,6 |
| IN BUF,\$\$LENG | EXIT |
| OUT BUF,\$\$LENG | HELLO DC 'Hello!' |
| OUT \$\$SIGN,6 | BILLY DC 'Billy' |
| EXIT | BUF DS 8 |
| \$\$TITLE DC 'Hello!' | END |
| \$\$SIGN DC 'Billy' | |
| BUF DS \$\$LENG | |
| END | |

実行例

〔プログラム〕

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

#define MCRDEF 20
#define STRLNG 32
#define LINLNG 256

int divide_line();
char linbuf[LINLNG], token[LINLNG][LINLNG],
                                delm[LINLNG];

char label[] = "$STRDEF";
char delmchar[] =
    " !\"#%&'()*+,-./:;<=>?[\\]^_`{|}~\n";

main() {
    FILE *sfp, *dfp;
    int mcrCnt = 0, tokCnt, flg, idx1, idx2;
    char orig[MCRDEF][STRLNG],
          expand[MCRDEF][STRLNG];

    sfp = fopen("source_file", "r");
    dfp = fopen("dest_file", "w");
    while(fgets(linbuf, LINLNG, sfp) != NULL) {
        tokCnt = divide_line();
        if(strcmp(token[0], label) == 0) {
            strcpy(orig[mcrCnt], token[1]);
            strcpy(expand[mcrCnt], token[2]);
            ;
        } else {
            for(idx1 = 0; idx1 < ; idx1++) {
                flg = 0;
                for(idx2 = 0; idx2 < mcrCnt; idx2++)
                    if(strcmp(token[idx1],
                               orig[idx2]) == 0) {
                        flg = 1;
                        break;
                    }
            }
            if()
                fprintf(dfp, "%s%c", token[idx1],
                        delm[idx1]);
            else
                fprintf(dfp, "%s%c", expand[idx2],
                        delm[idx1]);
        }
    }
    fclose(sfp);
    fclose(dfp);
}

int divide_line() {
    int tokCnt = 0, lidx = 0, tidx;

    do {
        for(tidx = 0; strchr(delmchar, linbuf[lidx])
            == NULL; lidx++, tidx++)
            token[tokCnt][tidx] = linbuf[lidx];
        token[tokCnt][tidx] = ;
        delm[tokCnt] = ;
        ;
    }
    tokCnt++;
}
} while(linbuf[lidx - 1] != '\n');
return tokCnt;
}
```

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 idx1 = 0 | イ idx1++ |
| ウ idx2 = 0 | エ idx2++ |
| オ mcrCnt = 0 | カ mcrCnt++ |
| キ tokCnt = 0 | ク tokCnt++ |

b に関する解答群

| | |
|----------|------------|
| ア idx2 | イ idx2++ |
| ウ mcrCnt | エ mcrCnt++ |
| オ tokCnt | カ tokCnt++ |

c に関する解答群

| | |
|-----------------|-----------------|
| ア flg = 0 | イ flg == 0 |
| ウ flg = 1 | エ flg == 1 |
| オ idx2 = 0 | カ idx2 > 0 |
| キ idx2 < mcrCnt | ク idx2 = mcrCnt |

d, e に関する解答群

| | |
|----------------|----------------|
| ア '\0' | イ '\n' |
| ウ delm[lidx] | エ delm[tidx] |
| オ linbuf[lidx] | カ linbuf[tidx] |
| キ token[lidx] | ク token[tidx] |

f に関する解答群

| | |
|--------------|------------|
| ア lidx = 0 | イ lidx++ |
| ウ tidx = 0 | エ tidx++ |
| オ tokCnt = 0 | カ tokCnt++ |

平成14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〕

(1) 自然数 num (num 2) を素因数に分解し、結果を次の例のように印字するプログラムである。

例 1 . 自然数 12 を入力した場合

数値 (2 以上の自然数) を入力してください。 : 12
12 = 2 × 2 × 3

例 2 . 自然数 3 を入力した場合

数値 (2 以上の自然数) を入力してください。 : 3
3 = 素数

(2) 素因数分解及び素数の判定手順は、次のとおりである。

変数 factor の値を 2 から始めて 1 ずつ増やしなが
ら、num を factor で割った結果が factor 以
上である間、 を実行する。

num が factor で割り切れる場合、次の処理を num
が factor で割り切れなくなるまで繰り返す。

- (a) factor の値を素因数として抽出する。
- (b) num を factor で割った商を新たな num とする。

最終的に割り切る factor がなかった場合には、
num は素数であると判定する。

(3) このプログラムでは、(2) の手順 での繰り返し処理の
回数を減らすために、まず最初に自然数 num が 2 の倍
数であるかどうかを判定する。num が 2 の倍数の場合
には、(2) の手順 によって、最初に 2 の因子をすべて
抽出する。続いて、変数 factor の値を、3 から始めて
2 ずつ増やしなが
ら、num を factor で割った結果が
factor 以上である間、素因数の抽出を繰り返す。

〔プログラム〕

(行番号)

```

1 #include <stdio.h>
2 #define TRUE 1
3 #define FALSE 0
4 main()
5 {
6     int num, prime = TRUE, factor = 2;
7     printf( "数値 ( 2 以上の自然数 ) を入力して
8         ください。 : " );
9     scanf( "%d", &num );
10    printf( "%d =", num );
11    if ( num != factor )
12        while ( num % factor == 0 ) { /*
13            数値が 2 の倍数 ? */
14                if ( prime == FALSE ) printf( " × " );
15                printf( " %d", factor );
16                a;
17                prime = FALSE;
18            }
19            factor++;
20            while ( b ) {
21                while ( num % factor == 0 ) {
22                    if ( prime == FALSE ) printf( " × " );

```

```

21         printf( " %d", factor );
22         a;
23         prime = FALSE;
24     }
25     c;
26 }
27 if ( prime == TRUE ) printf( " 素数\n" );
28 else if ( num > 1 ) printf
29     ( " × %d\n", d );
30 else printf( "\n" );
31 }

```

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, c に関する解答群

- | | |
|-----------------|-----------------|
| ア factor-- | イ factor += 2 |
| ウ num /= factor | エ num %= factor |
| オ prime = FALSE | カ prime = TRUE |

b に関する解答群

- ア num / factor >= factor
- イ num % factor != 0
- ウ prime == FALSE
- エ prime == TRUE

d に関する解答群

- | | |
|----------------|----------------|
| ア factor | イ num / factor |
| ウ num % factor | エ num |
| オ prime | |

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〕

画面上に凸四角形を描く関数 DrawRect を作成した。
関数 DrawRect は、次の大域変数を参照して動作する。

```

int Vx[4], Vy[4]; /* 4 頂点の座標値配列
                ( Vx は x 座標, Vy は y 座標 ) */

```

ただし、第 1 頂点 P1 から第 3 頂点 P3 までは、画面上
で時計回りになっており、その座標値は順番に Vx と Vy
に格納されている。画面上の座標系と原点の位置を図 1 に
示す。

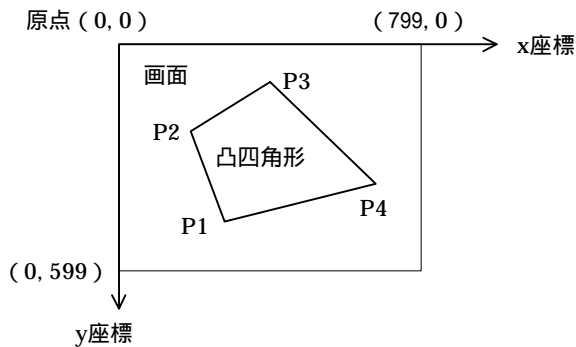


図1 画面上の座標系と原点の位置

第1頂点 P1 から第4頂点 P4 までは折れ線で結び、次に第4頂点 P4 の位置を検査し、凸四角形になる場合には P4 と P1 を結ぶ線分を描画して、返却値 0 を返す。4頂点を結ぶ図形が、凸四角形ではなく、図2に示すような図形となる場合には、P4 と P1 を結ぶ線分を描かずに、返却値 -1 を返す。

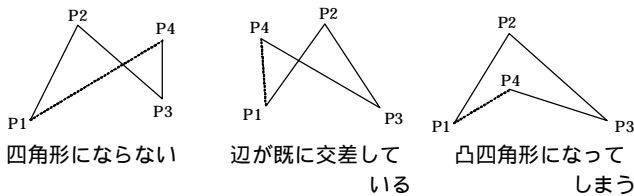


図2 凸四角形でない図形の例

なお、線分を引くために、次の二つの関数が用意されている。

```
void gMoveTo(int sx, int sy);
```

機能：線分の始点座標 (sx, sy) を設定する。この関数は描画を行わない。

```
void gLineTo(int ex, int ey);
```

機能：現在の始点から (ex, ey) まで線分を描き、次の描画のために始点を (ex, ey) に更新する。

〔プログラム〕

```
void gMoveTo(int, int);
void gLineTo(int, int);

int nCheck(double a)
{
    if (a > 0.0) return 1;
    else if (a < 0.0) return -1;
    return 0;
}

int DrawRect( )
{
```

```
int nCnt;
double dx, dy, k;
int nSide0, nSide1, nSide2;

for (nCnt = 0; nCnt < 4; nCnt++) {
    if (nCnt == 0) gMoveTo (Vx[nCnt], Vy[nCnt]);
    else if (nCnt < 3) gLineTo(Vx[nCnt], Vy[nCnt]);
    else {
        gLineTo(Vx[nCnt], Vy[nCnt]);

        /* P3 と P4 を通る直線の方程式を求める。 */
        dx = Vx[3] - Vx[2];
        dy = Vy[3] - Vy[2];
        k = dx * Vy[2] - dy * Vx[2];
        /* 第1の判定 */
        nSide0 = nCheck(dy * Vx[0] - dx * Vy[0] + k);
        nSide1 = nCheck(dy * Vx[1] - dx * Vy[1] + k);
        if (nSide0 != nSide1) return -1;
        /* P4 と P1 を通る直線の方程式を求める。 */
        dx = Vx[0] - Vx[3];
        dy = Vy[0] - Vy[3];
        k = dx * Vy[3] - dy * Vx[3];
        /* 第2の判定 */
        nSide1 = nCheck(dy * Vx[1] - dx * Vy[1] + k);
        nSide2 = nCheck(dy * Vx[2] - dx * Vy[2] + k);
        if (nSide1 != nSide2) return -1;
        /* P3 と P1 を通る直線の方程式を求める。 */
        dx = Vx[0] - Vx[2];
        dy = Vy[0] - Vy[2];
        k = dx * Vy[2] - dy * Vx[2];
        /* 第3の判定 */
        nSide1 = nCheck(dy * Vx[1] - dx * Vy[1] + k);
        nSide2 = nCheck(dy * Vx[3] - dx * Vy[3] + k);
        if (nSide1 * nSide2 != -1) return -1;
    }
}
gLineTo(Vx[0], Vy[0]);
return 0;
}
```

設問1 関数 DrawRect は、描く図形が凸四角形かどうか調べるため、直線の方程式を利用して辺の交差を判定している。この処理に関する次の記述中の に入れる正しい答えを、解答群の中から選べ。ただし、L12は、第1頂点 P1 と、第2頂点 P2 を結ぶ線分を表す。

2点 P3 (x₃, y₃), P4 (x₄, y₄) を通る直線の方程式を求めるとこの方程式は

$$(x_4 - x_3)(y - y_3) = (y_4 - y_3)(x - x_3)$$

と表すことができる。P4 と P3 の x 座標の差を dx,

y 座標の差を dy とすると、

$$dx = Vx[3] - Vx[2]$$

$$dy = Vy[3] - Vy[2]$$

である。この値を方程式に代入すると、

$$dx(y - y_3) = dy(x - x_3)$$

となる。この式を展開すると、

$$dx \times y - dx \times y_3 = dy \times x - dy \times x_3$$

となる。ここで、定数をまとめて k とすると、直線の方程式は、

$$dy \times x - dx \times y + k = 0$$

と表すことができる。

定数 k は、x と y に の座標値を代入すれば求めることができるので、求める直線の方程式は、 = 0 となる。

凸四角形になるためには、P3 と P4 を通る直線と、線分 L12 が交差してはならない。この交差を判定するため、 の x と y に、P1 と P2 の座標値を代入し、計算結果の を比較して判定する。

a, c に関する解答群

- | | |
|-----------|-----------|
| ア 第1頂点 P1 | イ 第2頂点 P2 |
| ウ 第3頂点 P3 | エ 絶対値 |
| オ 符号 | |

b に関する解答群

- | |
|---|
| ア $dx \times x ? dy \times y + dx \times Vy[2] ? dy \times Vx[2]$ |
| イ $dx \times x ? dy \times y + dy \times Vy[2] ? dx \times Vx[2]$ |
| ウ $dy \times x ? dx \times y + dx \times Vy[2] ? dy \times Vx[2]$ |
| エ $dy \times x ? dx \times y + dy \times Vx[2] ? dx \times Vy[2]$ |
| オ $dy \times x ? dx \times y + dy \times Vy[2] ? dx \times Vx[2]$ |

設問2 このプログラムは、描く図形が凸四角形になるかどうかを判定するため、三つの判定を順番に行っている。この判定のアルゴリズムに関する次の記述中の に入れる正しい答えを、解答群の中から選べ。

(第1の判定)

のような図形を無効にするため、P3 と P4 を通る直線の方程式を求めて、P1 と P2 の位置を検査する。

(第2の判定)

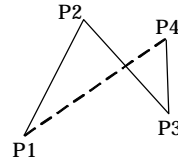
や のような図形を無効にするため、P4 と P1 を通る直線の方程式を求めて、P2 と P3 の位置を検査する。

(第3の判定)

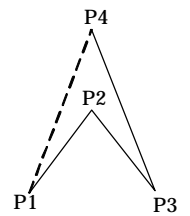
のような図形を無効にするため、P3 と P1 を通る直線の方程式を求めて、P2 と P4 の位置を検査する。

解答群

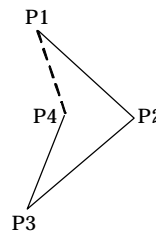
ア



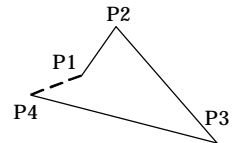
イ



ウ



エ



平成14年度 春期 FE 午後解答 C言語

問6

設問

a - オ b - エ c - イ

問10

設問

a - カ b - オ c - イ d - ア
e - オ f - イ

平成14年度 秋期 FE 午後解答 C言語

問6

設問

a - ウ b - ア c - イ d - エ

問10

設問1

a - ウ b - ウ c - オ

設問2

d - ウ e - ア f - エ g - イ (e, f 順不同)