問３　クラスタ分析に用いる k-means 法に関する次の記述を読んで，設問 1～3 に答えよ。

k-means 法によるクラスタ分析は，異なる性質のものが混ざり合った母集団から互いに似た性質をもつものを集め，クラスタと呼ばれる互いに素な部分集合に分類する手法である。新聞記事のビッグデータ検索，店舗の品ぞろえの分類，教師なし機械学習などで利用されている。ここでは，2 次元データを扱うこととする。

〔分類方法と例〕

N 個の点を K 個（N 未満）のクラスタに分類する方法を(1)～(5)に示す。

(1)　N 個の点（1 から N までの番号が付いている）からランダムに K 個の点を選び（以下，初期設定という），それらの点をコアと呼ぶ。コアには 1 から K までのコア番号を付ける。なお，K 個のコアの座標は全て異なっていなければならない。

(2)　N 個の点と K 個のコアとの距離をそれぞれ計算し，各点から見て，距離が最も短いコア（複数存在する場合は，番号が最も小さいコア）を選ぶ。選ばれたコアのコア番号を，各点が所属する 1 回目のクラスタ番号（1 から K）とする。ここで，二つの点を XY 座標を用いて P=(a，b)と Q=(c，d)とした場合，P と Q の距離を $\sqrt{(a-c)^2+(b-d)^2}$ で計算する。

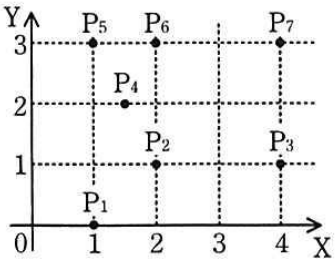
(3)　K 個のクラスタのそれぞれについて，クラスタに含まれる全ての点を使って重心を求める。ここで，重心の X 座標をクラスタに含まれる点の X 座標の平均，Y 座標をクラスタに含まれる点の Y 座標の平均と定義する。ここで求めた重心の番号はクラスタの番号と同じとする。

(4)　N 個の点と各クラスタの重心（G_1，…，G_K）との距離をそれぞれ計算し，各点から見て距離が最も短い重心（複数存在する場合は，番号が最も小さい重心）を選ぶ。選ばれた重心の番号を，各点が所属する次のクラスタ番号とする。

(5)　重心の座標が変わらなくなるまで，(3)と(4)を繰り返す。

次の座標で与えられる 7 個の点を，この分類方法に従い，二つのクラスタに分類する例を図 1 に示す。

$P_1=(1, 0)$　$P_2=(2, 1)$　$P_3=(4, 1)$　$P_4=(1.5, 2)$　$P_5=(1, 3)$　$P_6=(2, 3)$　$P_7=(4, 3)$



(1) 初期設定はコア 1＝P_2，コア 2＝P_5とする。
(2) 各点とコアの距離から表 1 のように所属クラスタを求める。
(3) クラスタの重心を計算し，クラスタ 1 の重心 G_1は（2.5，1.4），クラスタ 2 の重心 G_2は［　ア　］となる。
(4) 各点と重心の距離から表 2 のように所属クラスタを見直す。
(5) (3)と(4)を繰り返し，重心の座標が変わらなくなると①分類が完了する。

図 1　7 個の点の分類

表 1　コアとの距離と所属クラスタ

点	コア 1 との距離	コア 2 との距離	所属クラスタ番号
P_1	$\sqrt{2}$	3	1
P_2	0	$\sqrt{5}$	1
P_3	2	$\sqrt{13}$	1
P_4	$\sqrt{5}/2$	$\sqrt{5}/2$	1
P_5	$\sqrt{5}$	0	2
P_6	2	1	2
P_7	$2\sqrt{2}$	3	1

表 2　重心との距離と次の所属クラスタ

点	重心 G_1 との距離	重心 G_2 との距離	次の所属クラスタ番号
P_1	2.05	3.04	1
P_2	0.64	2.06	1
P_3	1.55	3.20	1
P_4	1.16	1.00	2
P_5	2.19	0.50	2
P_6	1.67	0.50	2
P_7	2.19	2.50	1

注記　距離は小数第 3 位以下切捨て

〔クラスタ分析のプログラム〕

この手法のプログラムを図 2 に，プログラムで使う主な変数，関数及び配列を表 3 に示す。ここで，配列の添字は全て 1 から始まり，要素の初期値は全て 0 とする。

表 3　クラスタ分析のプログラムで使う主な変数，関数及び配列

名称	種類	内容
core[t]	配列	コア番号が t である点 P_mの番号 m を格納する。
initial(K)	関数	初期設定として次の処理を行う。N 個の点 {P_1, P_2, …, P_N} からランダムに異なる K 個を抽出し，その番号を順に配列 core に格納する。
data_dist(s, t)	関数	点 P_sとコア番号が t である点の距離を返す。
grav_dist(s, t)	関数	点 P_sと重心 G_tの距離を返す。
data_length[t]	配列	ある点から見た，コア番号が t である点との距離を格納する。
grav_length[t]	配列	ある点から見た，重心 G_tとの距離を格納する。
min_index( ) 引数は data_length 又は grav_length	関数	配列の中で，最小値が格納されている添字 s を返す。最小値が二つ以上ある場合は，最も小さい添字を返す。 例　右の配列に対する戻り値は2 添字　1 2 3 4 5 配列　5 1 4 1 3
cluster[s]	配列	点 P_sが所属するクラスタ番号を格納する。
coordinate_x[s]	配列	重心 G_sの X 座標を格納する。

表３　クラスタ分析のプログラムで使う主な変数，関数及び配列（続き）

coordinate_y[s]	配列	重心 G_s の Y 座標を格納する。
gravity_x(s)	関数	クラスタ s の重心 G_s を求め，その X 座標を返す。
gravity_y(s)	関数	クラスタ s の重心 G_s を求め，その Y 座標を返す。
flag	変数	フラグ（値は 0 又は 1）

```
function clustering (N, K)                    //N はデータ数，K はクラスタ数
    MaxCount ← 100                            //無限ループを防ぐため最大見直し回数を設定
    initial(K)                                //初期設定
    for( s を 1 から N まで 1 ずつ増やす )     //クラスタを決定（1 回目）
        for( t を 1 から K まで 1 ずつ増やす )
            data_length[t] ← data_dist(s, core[t])
        endfor
        cluster[s] ← min_index(data_length)
    endfor
    for( p を 1 から MaxCount まで 1 ずつ増やす )     //クラスタを見直す
        if( p が 1 と等しい )
            for( [   イ   ] )                 //1 回目の重心の計算
                coordinate_x[t] ← gravity_x(t)
                coordinate_y[t] ← gravity_y(t)
            endfor
        else
            [   ウ   ]
            for( [   イ   ] )                 //2 回目以降の重心の計算
                if( gravity_x(t)と coordinate_x[t]が異なる     //重心を修正する
                    又は gravity_y(t)と coordinate_y[t]が異なる )
                      coordinate_x[t] ← gravity_x(t)
                      coordinate_y[t] ← gravity_y(t)
                      flag ← 1
                endif
            endfor
            if( [   エ   ] )                  //終了して抜ける
                return
            endif
        endif
        for( s を 1 から N まで 1 ずつ増やす )       //近い重心を見つける
            for( t を 1 から K まで 1 ずつ増やす )
                grav_length[t] ← grav_dist(s,t)
            endfor
            [          オ          ]
        endfor
    endfor
endfunction
```

図２　クラスタ分析の関数 clustering のプログラム

〔初期設定の改良〕

このアルゴリズムの最終結果は初期設定に依存し，そこでのコア間の距離が短いと適切な分類結果を得にくい。そこで，関数 initial において一つ目のコアはランダムに選び，それ以降はコア間の距離が長くなる点が選ばれやすくなるアルゴリズムを検討した。検討したアルゴリズムでは，t 番目までのコアが決まった後，t+1 番目のコアを残った点から選ぶときに，それまでに決まったコアから離れた点を，より高い確率で選ぶようにする。具体的には，それまでに決まったコア（コア 1～コア t）と，残った N−t 個の点から選んだ点 P_s との距離の和を T_s とする。N−t 個の全ての点について T_s を求め，T_s が［　カ　］ほど高い確率で点 P_s が選ばれるようにする。このとき，点 P_s が t+1 番目のコアとして選ばれる確率として［　キ　］を適用する。

設問１　〔分類方法と例〕について，(1)，(2)に答えよ。

(1)　図１中の［　ア　］に入れる座標を答えよ。

(2)　図 1 中の下線①のように分類が完了したときに，P_1 と同じクラスタに入る点を全て答えよ。

設問２　図２中の［　イ　］～［　オ　］に入れる適切な字句を答えよ。

設問３　〔初期設定の改良〕について，(1)，(2)に答えよ。

(1)　本文中の［　カ　］に入れる適切な字句を解答群の中から選び，記号で答えよ。

解答群

ア　大きい　　イ　小さい

(2)　本文中の［　キ　］に入れる適切な式を T_s と Sum を使って答えよ。

ここで，Sum は N−t 個の全ての T_s の和とする。