問8　クイックソートを応用した選択アルゴリズム（データ構造及びアルゴリズム）（H27春-FE 午後問8）

【解答】

［設問1］　a－ア，b－ウ
［設問2］　c－イ，d－エ
［設問3］　e－オ，f－エ

【解説】

クイックソートを応用した選択アルゴリズムに関する問題である。設問は，全てプログラムのトレース（追跡）に関する内容であり，配列や変数の変化を図で表して丁寧にトレースすることが必要である。ソートのアルゴリズムは午前問題でも問われる内容であり，クイックソートがどのようにデータを整列するのかを理解できていると，プログラムの処理内容を理解しやすかったと思われる。ソートや探索のアルゴリズムは問8でよく扱われるテーマであり，各アルゴリズムがどのように処理を行うかをしっかり理解しておきたい。また，処理のトレースについても，よく練習しておきたい。

はじめに，クイックソートではどのように処理を行うのかについて，簡単に確認する。クイックソートは，基準となる値（ピボットという）を決め，基準値よりも小さい値のグループと基準値よりも大きい値のグループに分割し，更に分割したグループで基準値を決め，基準値よりも小さい値のグループと大きい値のグループに分割する。この処理をグループ内のデータ数が1になるまで繰り返すことでソートを行うアルゴリズムである。具体的には次のように処理を行う。

基準値を5として基準値よりも小さい値と大きい値のグループに分割する

6	3	1	10	5	7	4	8	9	2

分割した各グループを更に分割する（基準値は3と8）

2	3	1	4		5		7	10	8	9	6

※網掛け部分は基準値だった値

同様にデータが2個以上あるグループを更に分割する

2	1		3		4		5		7	6		8		9	10

全てのグループのデータが1個になるとソート完了

1		2		3		4		5		6		7		8		9		10

続いて，関数 Select の処理を確認する。

〔プログラム〕
（行番号）

```
 1  ○整数型: Select(整数型: x[], 整数型: n, 整数型: k)
 2  ○整数型: Top, Last, Pivot, i, j, work

 3   ・Top ← 1                    /* 走査範囲の左端の初期値を設定 */
 4   ・Last ← n                   /* 走査範囲の右端の初期値を設定 */
 5  ■ Top < Last
 6   | ・Pivot ← x[k]
 7   | ・i ← Top                  ←――― α
 8   | ・j ← Last
 9   | ■ true                     /* ループ */
10   |  | ■ x[i] < Pivot         ←――― β
11   |  | ・i ← i+1
12   |  |
13   |  | ■ Pivot < x[j]
14   |  | ・j ← j−1
15   |  |
16   |  | ▲ i ≧ j
17   |  | ・break                /* ループから抜ける */
18   |  ▼
19   |  | ・work ← x[i]
20   |  | ・x[i] ← x[j]
21   |  | ・x[j] ← work          ←――― γ
22   |  | ・i ← i+1
23   |  | ・j ← j−1
24   | ■
25   | ▲ i ≦ k
26   | ・Top← j+1
27   | ▼
28   | ▲ k ≦ j
29   | ・Last ←i−1
30   | ▼
31  ■
32   ・return x[k]
```

行番号1：
　関数（手続）の宣言。Select は関数名，（　）の中は，関数が受け取る値（引数）を格納する変数名と型の宣言。関数名の前に記述した型は，関数の返却値の型を表す。これらについては，〔関数 Select の引数／返却値の仕様〕で示されている。
行番号2：
　関数内で使用する変数と型の宣言。
行番号3，4：
　配列を走査する範囲の左端と右端の要素番号を設定。最初の走査範囲は配列全体であるため，Top（左端）には1，Last（右端）には n を設定する。
行番号5～31：

k 番目に小さい値を探す処理。走査範囲に含まれる要素数が1以下になるまで繰り返す。
行番号6～8：
　グループ分けの基準値 Pivot，配列 x の添字として使用する i と j に走査範囲を表す Top と Last を設定。
行番号9～24：
　基準値 Pivot 以下の値と基準値以上の値に分ける処理。繰返し条件に true（真）を設定することで無限ループとなる。行番号16の条件式が成り立つ場合に実行される行番号17の break によってこのループから抜け，行番号25以降の処理を実行する。
行番号10～12：
　基準値 Pivot 以上の値の検索。配列 x の要素が Pivot よりも小さい間，処理を繰り返す。
行番号13～15：
　基準値 Pivot 以下の値の検索。配列 x の要素が Pivot よりも大きい間，処理を繰り返す。
行番号16～18：
　この条件が成り立つと，行番号9～24のループを抜ける。i≧j は，現在の走査範囲のグループ分けが終了したことを意味する。
行番号19～21：
　行番号10～12の繰返し処理で見つかった基準値 Pivot 以上の値と，行番号13～15の繰返し処理で見つかった基準値 Pivot 以下の値を入れ替える処理。
行番号22，23：
　変数 i と j の更新。次に参照する要素を一つ後ろと一つ前にずらす。
行番号25～30：
　求める値（k 番目に小さい値）が含まれるグループを確定。
　行番号25～27：走査範囲の左端を設定。
　行番号28～30：走査範囲の右端を設定。
行番号32：
　戻り値（k 番目に小さい値）の返却。

［設問1］
・空欄 a，b：与えられた値（k は3，Top は1，Last は7）でトレースすると次のようになる。なお，配列 x の内容で網掛けされた値は Pivot と比較する要素，囲み線つきの値は入れ替えた要素を表す。

Pivot	i	j	配列 x の内容	処理
6	1	7	3, 5, 6, 4, 7, 2, 1	x[i]<Pivot を満たす i に+1する
6	2	7	3, 5, 6, 4, 7, 2, 1	x[i]<Pivot を満たす i を+1する
6	3	7	3, 5, 6, 4, 7, 2, 1	x[i]<Pivot を満たさない 行番号10～12のループを抜ける
6	3	7	3, 5, 6, 4, 7, 2, 1	x[j]>Pivot を満たさない 行番号13～15のループを抜ける
6	3	7	3, 5, 1, 4, 7, 2, 6	i≧j を満たさない 行番号9～24のループは抜けない x[i]と x[j]の要素を入れ替える i を+1，j を−1する 行番号9に戻る
6	4	6	3, 5, 1, 4, 7, 2, 6	x[i]<Pivot を満たす i を+1する
6	5	6	3, 5, 1, 4, 7, 2, 6	x[i]<Pivot を満たさない 行番号10～12のループを抜ける
6	5	6	3, 5, 1, 4, 7, 2, 6	x[j]>Pivot を満たさない 行番号13～15のループを抜ける
6	5	6	3, 5, 1, 4, 2, 7, 6	i≧j を満たさない 行番号9～24のループは抜けない x[i]と x[j]の要素を入れ替える i を+1，j を−1する 行番号9に戻る
6	6	5	3, 5, 1, 4, 2, 7, 6	x[i]<Pivot を満たさない 行番号10～12のループを抜ける
6	6	5	3, 5, 1, 4, 2, 7, 6	x[j]>Pivot を満たさない 行番号13～15のループを抜ける
6	6	5	3, 5, 1, 4, 2, 7, 6	i≧j を満たす 行番号9～24のループを抜ける i≦k を満たさない Top の値は変更なし（Top=1） k≦j を満たす i−1を Last に格納する（Last=5） 行番号5に戻る

以上より，Top には1，Last には5が設定されるため，空欄 a は（ア）が正解である。

続いて2回目の選択処理をトレースする。なお，k は3，Top は1，Last は5であり，行番号6～8の処理によって，Pivot は1，i は1，j は5に更新される。

Pivot	i	j	配列 x の内容	処理
1	1	5	3, 5, 1, 4, 2, 7, 6	x[i]<Pivot を満たさない 行番号10～12のループを抜ける
1	1	5	3, 5, 1, 4, 2, 7, 6	Pivot<x[j]を満たす j を−1する
1	1	4	3, 5, 1, 4, 2, 7, 6	Pivot<x[j]を満たす j を−1する
1	1	3	3, 5, 1, 4, 2, 7, 6	Pivot<x[j]を満たさない 行番号13～15のループを抜ける
1	1	3	1, 5, 3, 4, 2, 7, 6	i≧j を満たさない 行番号9～24のループは抜けない x[i]と x[j]の要素を入れ替える i を+1，j を−1する 行番号9に戻る
1	2	2	1, 5, 3, 4, 2, 7, 6	x[i]<Pivot を満たさない 行番号10～12のループを抜ける
1	2	2	1, 5, 3, 4, 2, 7, 6	Pivot<x[j]を満たす j を−1する
1	2	1	1, 5, 3, 4, 2, 7, 6	Pivot<x[j]を満たさない 行番号13～15のループを抜ける
				i≧j を満たす 行番号9～24のループを抜ける i≦k を満たす

問8　クイックソートを応用した選択アルゴリズム（データ構造及びアルゴリズム）（H27春-FE 午後問8）

【解答】

［設問1］　a－ア，b－ウ
［設問2］　c－イ，d－エ
［設問3］　e－オ，f－エ

【解説】

クイックソートを応用した選択アルゴリズムに関する問題である。設問は，全てプログラムのトレース（追跡）に関する内容であり，配列や変数の変化を図で表して丁寧にトレースすることが必要である。ソートのアルゴリズムは午前問題でも問われる内容であり，クイックソートがどのようにデータを整列するのかを理解できていると，プログラムの処理内容を理解しやすかったと思われる。ソートや探索のアルゴリズムは問8でよく扱われるテーマであり，各アルゴリズムがどのように処理を行うかをしっかり理解しておきたい。また，処理のトレースについても，よく練習しておきたい。

はじめに，クイックソートではどのように処理を行うのかについて，簡単に確認する。クイックソートは，基準となる値（ピボットという）を決め，基準値よりも小さい値のグループと基準値よりも大きい値のグループに分割し，更に分割したグループで基準値を決め，基準値よりも小さい値のグループと大きい値のグループに分割する。この処理をグループ内のデータ数が1になるまで繰り返すことでソートを行うアルゴリズムである。具体的には次のように処理を行う。

基準値を5として基準値よりも小さい値と大きい値のグループに分割する

6	3	1	10	5	7	4	8	9	2

分割した各グループを更に分割する（基準値は3と8）

2	3	1	4		5		7	10	8	9	6

※網掛け部分は基準値だった値

同様にデータが2個以上あるグループを更に分割する

2	1		3		4		5		7	6		8		9	10

全てのグループのデータが1個になるとソート完了

1		2		3		4		5		6		7		8		9		10

続いて，関数 Select の処理を確認する。

〔プログラム〕
（行番号）

```
 1  ○整数型: Select(整数型: x[], 整数型: n, 整数型: k)
 2  ○整数型: Top, Last, Pivot, i, j, work

 3   ・Top ← 1                    /* 走査範囲の左端の初期値を設定 */
 4   ・Last ← n                   /* 走査範囲の右端の初期値を設定 */
 5  ■ Top < Last
 6   | ・Pivot ← x[k]
 7   | ・i ← Top                  ←――― α
 8   | ・j ← Last
 9   | ■ true                     /* ループ */
10   |  | ■ x[i] < Pivot         ←――― β
11   |  | ・i ← i+1
12   |  |
13   |  | ■ Pivot < x[j]
14   |  | ・j ← j−1
15   |  |
16   |  | ▲ i ≧ j
17   |  | ・break                /* ループから抜ける */
18   |  ▼
19   |  | ・work ← x[i]
20   |  | ・x[i] ← x[j]
21   |  | ・x[j] ← work          ←――― γ
22   |  | ・i ← i+1
23   |  | ・j ← j−1
24   | ■
25   | ▲ i ≦ k
26   | ・Top← j+1
27   | ▼
28   | ▲ k ≦ j
29   | ・Last ←i−1
30   | ▼
31  ■
32   ・return x[k]
```

行番号1：
　関数（手続）の宣言。Select は関数名，（　）の中は，関数が受け取る値（引数）を格納する変数名と型の宣言。関数名の前に記述した型は，関数の返却値の型を表す。これらについては，〔関数 Select の引数／返却値の仕様〕で示されている。
行番号2：
　関数内で使用する変数と型の宣言。
行番号3，4：
　配列を走査する範囲の左端と右端の要素番号を設定。最初の走査範囲は配列全体であるため，Top（左端）には1，Last（右端）には n を設定する。
行番号5～31：

k 番目に小さい値を探す処理。走査範囲に含まれる要素数が1以下になるまで繰り返す。
行番号6～8：
　グループ分けの基準値 Pivot，配列 x の添字として使用する i と j に走査範囲を表す Top と Last を設定。
行番号9～24：
　基準値 Pivot 以下の値と基準値以上の値に分ける処理。繰返し条件に true（真）を設定することで無限ループとなる。行番号16の条件式が成り立つ場合に実行される行番号17の break によってこのループから抜け，行番号25以降の処理を実行する。
行番号10～12：
　基準値 Pivot 以上の値の検索。配列 x の要素が Pivot よりも小さい間，処理を繰り返す。
行番号13～15：
　基準値 Pivot 以下の値の検索。配列 x の要素が Pivot よりも大きい間，処理を繰り返す。
行番号16～18：
　この条件が成り立つと，行番号9～24のループを抜ける。i≧j は，現在の走査範囲のグループ分けが終了したことを意味する。
行番号19～21：
　行番号10～12の繰返し処理で見つかった基準値 Pivot 以上の値と，行番号13～15の繰返し処理で見つかった基準値 Pivot 以下の値を入れ替える処理。
行番号22，23：
　変数 i と j の更新。次に参照する要素を一つ後ろと一つ前にずらす。
行番号25～30：
　求める値（k 番目に小さい値）が含まれるグループを確定。
　行番号25～27：走査範囲の左端を設定。
　行番号28～30：走査範囲の右端を設定。
行番号32：
　戻り値（k 番目に小さい値）の返却。

［設問1］
・空欄 a，b：与えられた値（k は3，Top は1，Last は7）でトレースすると次のようになる。なお，配列 x の内容で網掛けされた値は Pivot と比較する要素，囲み線つきの値は入れ替えた要素を表す。

Pivot	i	j	配列 x の内容	処理
6	1	7	3, 5, 6, 4, 7, 2, 1	x[i]<Pivot を満たす i に+1する
6	2	7	3, 5, 6, 4, 7, 2, 1	x[i]<Pivot を満たす i を+1する
6	3	7	3, 5, 6, 4, 7, 2, 1	x[i]<Pivot を満たさない 行番号10～12のループを抜ける
6	3	7	3, 5, 6, 4, 7, 2, 1	x[j]>Pivot を満たさない 行番号13～15のループを抜ける
6	3	7	3, 5, 1, 4, 7, 2, 6	i≧j を満たさない 行番号9～24のループは抜けない x[i]と x[j]の要素を入れ替える i を+1，j を−1する 行番号9に戻る
6	4	6	3, 5, 1, 4, 7, 2, 6	x[i]<Pivot を満たす i を+1する
6	5	6	3, 5, 1, 4, 7, 2, 6	x[i]<Pivot を満たさない 行番号10～12のループを抜ける
6	5	6	3, 5, 1, 4, 7, 2, 6	x[j]>Pivot を満たさない 行番号13～15のループを抜ける
6	5	6	3, 5, 1, 4, 2, 7, 6	i≧j を満たさない 行番号9～24のループは抜けない x[i]と x[j]の要素を入れ替える i を+1，j を−1する 行番号9に戻る
6	6	5	3, 5, 1, 4, 2, 7, 6	x[i]<Pivot を満たさない 行番号10～12のループを抜ける
6	6	5	3, 5, 1, 4, 2, 7, 6	x[j]>Pivot を満たさない 行番号13～15のループを抜ける
6	6	5	3, 5, 1, 4, 2, 7, 6	i≧j を満たす 行番号9～24のループを抜ける i≦k を満たさない Top の値は変更なし（Top=1） k≦j を満たす i−1を Last に格納する（Last=5） 行番号5に戻る

以上より，Top には1，Last には5が設定されるため，空欄 a は（ア）が正解である。

続いて2回目の選択処理をトレースする。なお，k は3，Top は1，Last は5であり，行番号6～8の処理によって，Pivot は1，i は1，j は5に更新される。

Pivot	i	j	配列 x の内容	処理
1	1	5	3, 5, 1, 4, 2, 7, 6	x[i]<Pivot を満たさない 行番号10～12のループを抜ける
1	1	5	3, 5, 1, 4, 2, 7, 6	Pivot<x[j]を満たす j を−1する
1	1	4	3, 5, 1, 4, 2, 7, 6	Pivot<x[j]を満たす j を−1する
1	1	3	3, 5, 1, 4, 2, 7, 6	Pivot<x[j]を満たさない 行番号13～15のループを抜ける
1	1	3	1, 5, 3, 4, 2, 7, 6	i≧j を満たさない 行番号9～24のループは抜けない x[i]と x[j]の要素を入れ替える i を+1，j を−1する 行番号9に戻る
1	2	2	1, 5, 3, 4, 2, 7, 6	x[i]<Pivot を満たさない 行番号10～12のループを抜ける
1	2	2	1, 5, 3, 4, 2, 7, 6	Pivot<x[j]を満たす j を−1する
1	2	1	1, 5, 3, 4, 2, 7, 6	Pivot<x[j]を満たさない 行番号13～15のループを抜ける
				i≧j を満たす 行番号9～24のループを抜ける i≦k を満たす