

かねじゅん模試

アルゴリズムとプログラミング

作成者はこの模試に関する一切の責任を負わない

次ページに示すプログラムは、配列 data 内の数値を昇順に整列する C 言語のプログラムである。関数 swap は、引数で指定された配列内の 2 つのデータを入れ替える処理を、関数 sort は配列内の数値を昇順に整列する処理を行う。なお、プログラムの行頭の数字は行番号を表す。

- (1) 関数 sort は、ある整列アルゴリズムのかねじゅんによる改造版である。元の整列アルゴリズムは一般に何と呼ばれているか答えよ。
- (2) この整列アルゴリズムは、元のアルゴリズムとどのような点が異なるか答えよ。
- (3) 空欄(ア)から(ウ)を適切に埋めよ。
- (4) この整列アルゴリズムが安定か安定でないか答えよ。また、その理由も答えよ。
- (5) 12 行目の for 文の 3 回目までの各ループにおける、19 行目実行時点での変数 min と max の値をそれぞれ答えよ。
- (6) 15 行目の for 文の処理内容 (16 行目と 17 行目) は、このプログラムが終了するまでに何回実行されるか答えよ。
- (7) 19 行目から 27 行目では、変数 left、right、min、max の保持する値の関係によってデータの交換順序を変えている。仮に、この部分の処理を 25 行目、26 行目のようなデータ交換順序のみとした場合データの整列が正しく行われない。この理由を、具体例を挙げて説明せよ。

```

1  ( ア ) <stdio.h>
2  #define N 20
3
4  void swap(int data[], int a, int b){
5      if(a != b){
6          int tmp = data[a]; data[a] = data[b]; data[b] = tmp;
7      }
8  }
9
10 void sort(int data[]){
11     int left, right, i, min, max;
12     for(( イ ); left < right; left++, right--){
13         min = left;
14         max = right;
15         for(i = 1; i <= right-left; i++){
16             if(data[left+i] < data[min]) min = left+i;
17             if(data[right-i] > data[max]) max = right-i;
18         }
19         if(left == max && right == min){
20             swap(data, left, right);
21         }else if(( ウ )){
22             swap(data, right, max);
23             swap(data, left, min);
24         }else{
25             swap(data, left, min);
26             swap(data, right, max);
27         }
28     }
29 }
30
31 int main(){
32     int data[N] = {12, 3, 8, 2, 17, 7, 9, 10, 1, 5, 16, 19, 13, 4, 15, 6, 14, 18, 20, 11};
33     sort(data);
34     return 0;
35 }

```

解答

- (1) 選択ソート
- (2) 1 回の配列の検査で最大値だけでなく最小値も発見し、データを入れ替える。
- (3) (ア) #include (イ) left = 0, right = N-1 (ウ) left == max
- (4) 安定でない。元のアルゴリズムである選択ソートが安定でない。データの交換の際に元の順序が失われる。(過去問ばかりやっていると不等号に気を奪われて安定と答えるであろう。)
- (5) 1 回目 min 8, max 18
2 回目 min 3, max 11
3 回目 min 3, max 17
- (6) $\sum_{i=0}^{\frac{N}{2}-1} (N-1-2i) = (N-1) * \left(\frac{N}{2}-1+1\right) - 2 * \frac{1}{2} \left(\frac{N}{2}-1\right) \left(\frac{N}{2}-1+1\right) = \frac{1}{4} N^2 = 100$
- (7) 例えば、left == max && right == min のとき、25 行目、26 行目の交換方法では、同じ 2 つのデータの交換が 2 回行われ、交換前と同じ状態になる。他にも、left == max のとき、はじめに left と min の交換をすると、後の right と max の交換の際に交換したかった、max (left) の値が変わっているため、意図した交換が行われない。