

2015 概論2 第2 順位

main関数内に、int型の2次元配列data[10][10]があります。各要素(計100個)には、それぞれ無作為に0～100から選ばれた整数が入っています。ある要素が、その要素と隣接する要素(上下／左右／斜めの最大8個、最小3個)と大きさを比べて、何番目に大きい値であるかを調べて、表示するプログラムを作成します。同じ値の順位は、順位は下がらないものとします(n位タイはn位とします)。サンプルプログラムを参考に、プログラムを作成しなさい。(なお、サンプルプログラムは解答中に記入してある)

<2次元配列dataへの入力例>

41	85	72	38	80	69	65	68	96	22
49	67	51	61	63	87	66	24	80	83
71	60	64	52	90	60	49	31	23	99
94	11	25	24	51	15	13	39	67	97
19	76	12	33	99	18	92	35	74	0
95	71	39	33	39	32	37	45	57	71
95	5	71	24	86	8	51	54	74	24
75	70	33	63	29	99	58	94	52	13
35	99	46	57	71	23	17	3	94	48
77	18	83	11	83	25	59	62	2	78

<2次元配列dataへ入力した値の大きさの順位例>

4	1	2	6	2	3	5	3	1	4
5	4	8	6	5	2	4	8	4	3
2	5	2	5	1	5	4	6	9	1
1	9	6	8	5	8	9	5	4	2
5	3	8	5	1	7	1	8	2	6
1	4	4	6	3	7	5	7	4	3
1	9	1	9	2	9	5	5	2	5
3	5	7	4	6	1	3	1	5	6
5	1	6	5	3	8	8	8	1	4
2	6	2	6	1	4	2	2	6	2

<初期ソースコード>

```
#include <iostream>
#include <stdlib.h>
using namespace std;

int main(){
    int data[10][10]; //データ用配列
    int tmp[10][10]; //データのコピー用配列
    //初期データをdataに入力しtmpにコピーする
    for (int i = 0; i < 10; i++){
        for (int j = 0; j < 10; j++){
            data[i][j] = rand() % 101;
            tmp[i][j] = data[i][j];
        }
    }
    //確認するために初期値を画面表示する
    for (int i = 0; i < 10; i++){
        for (int j = 0; j < 10; j++){
            cout << data[i][j] << "  ";
        }
        cout << endl;
    }
    cout << endl; //順位表示と区別するために空白行を挿入

    //以下にプログラムを記入しなさい

    //確認するために順位を画面表示する
    for (int i = 0; i < 10; i++){
        for (int j = 0; j < 10; j++){
            cout << data[i][j] << "  ";
        }
        cout << endl;
    }
    return 0;
}
```

<回答ソースコード例>

```
#include <iostream>
#include <stdlib.h>
using namespace std;

int main(){
    int data[10][10]; //データ用配列
    int tmp[10][10]; //データのコピー用配列

    //初期データをdataに入力しtmpにコピーする
    for (int i = 0; i < 10; i++){
        for (int j = 0; j < 10; j++){
            data[i][j] = rand() % 101;
            tmp[i][j] = data[i][j];
        }
    }

    //確認するために初期値を画面表示する
    for (int i = 0; i < 10; i++){
        for (int j = 0; j < 10; j++){
            cout << data[i][j] << " ";
        }
        cout << endl;
    }
    cout << endl; //順位表示と区別するために空白行を挿入

    // 各要素の順位を計算して data 配列に格納する

    // 各要素の順位を計算して data 配列に格納する
    for (int i = 0; i < 10; i++){
        for (int j = 0; j < 10; j++){
            int rank = 1; // 順位の初期値を1とする
            // 隣接要素と比較する
            for (int x = i - 1; x <= i + 1; x++){
                for (int y = j - 1; y <= j + 1; y++){
                    // 配列の範囲外は無視
                    if (x >= 0 && x < 10 && y >= 0 && y < 10){
                        if (x != i || y != j) { // 自分は無視
                            if (tmp[x][y] > tmp[i][j]){
                                rank++;
                            }
                        }
                    }
                }
            }
            data[i][j] = rank;
        }
    }

    //確認するために順位を画面表示する
    for (int i = 0; i < 10; i++){
        for (int j = 0; j < 10; j++){
```

```
        cout << data[i][j] << "  ";  
    }  
    cout << endl;  
}  
return 0;  
}
```

2017 概論1 第2中間(改) function.cpp

ユーザが2以上の任意の自然数を入力します。入力された値を素因数分解し、全ての素因数を小さい順に表示するプログラムを作成しなさい。ただし、以下のメイン関数を変更せずに用いること。また、用いるサブルーチンは「void function(int x)」の一つのみとし、このサブルーチンを再起呼び出して用いること。

出力例(1)

```
2以上の値を入力 :  
12  
入力値12の約数を昇順に表示 : 2, 2, 3,
```

出力例(2)

```
2以上の値を入力 :  
32  
入力値32の約数を昇順に表示 : 2, 2, 2, 2,
```

出力例(3)

```
2以上の値を入力 :  
71  
入力値71の約数を昇順に表示 : 71,
```

＜初期ソースコード＞

```
#include <cstdio>
#include <iostream>
using namespace std;

void function(int x) {
}

int main(){
    int n;
    cout << "2以上の値を入力:" << endl;
    cin >> n;
    cout << "入力値" << n << "の約数を昇順に表示:";
    function(n);
}
```

＜回答ソースコード例＞

```
#include <cstdio>
#include <iostream>
using namespace std;

void function(int x) {
    for (int i = 2; i <= x; i++) {
        if (x % i == 0) {
            cout << i << ",";
            function(x / i);
            return;
        }
    }
}

int main() {
    int n;
    cout << "2以上の値を入力:" << endl;
    cin >> n;
    cout << "入力値" << n << "の約数を昇順に表示:";
    function(n);
}
```

2018 概論2 第1中間 雨のしずく

n 個($n \leq 100$)の雨のしずくが生成された時刻($tc[n]$)と、雨のしずくが着地した時刻($td[n]$)が記録されたデータがある。

雨のしずく i は、生成時刻($tc[i]$)に生成され、着地時刻($td[i]$)に地面に到着し、消失する。

生成時刻($tc[i]$)、着地時刻($td[i]$)はそれぞれ、開始から経過した秒数である。

このような n 個のデータを集計し、雨のしずくが最も多く存在していた時の雨のしずくの個数 max_count を出力せよ。

なお、ここでは生成時刻($tc[i]$)／着地時刻($td[i]$)にも、雨のしずく i が存在していたとみなす。

また、すべての雨のしずく i において、 $0 \leq tc[i] \leq td[i] < 1000$ であるとする。

出力例(1)

雨のしずくの数(上限100)

4

0番目のしずくの生成時刻は？

0

0番目のしずくの着地時刻は？

31

1番目のしずくの生成時刻は？

1

1番目のしずくの着地時刻は？

41

2番目のしずくの生成時刻は？

3

2番目のしずくの着地時刻は？

51

3番目のしずくの生成時刻は？

2

3番目のしずくの着地時刻は？

20

4

出力例(2)

```
雨のしずくの数(上限100)
3
0番目のしずくの生成時刻は？
5
0番目のしずくの着地時刻は？
10
1番目のしずくの生成時刻は？
10
1番目のしずくの着地時刻は？
25
2番目のしずくの生成時刻は？
26
2番目のしずくの着地時刻は？
30
2
```

出力例(3)

```
雨のしずくの数(上限100)
10
0番目のしずくの生成時刻は？
84
0番目のしずくの着地時刻は？
88
1番目のしずくの生成時刻は？
56
1番目のしずくの着地時刻は？
70
2番目のしずくの生成時刻は？
39
2番目のしずくの着地時刻は？
100
3番目のしずくの生成時刻は？
74
3番目のしずくの着地時刻は？
77
4番目のしずくの生成時刻は？
48
4番目のしずくの着地時刻は？
73
5番目のしずくの生成時刻は？
8
5番目のしずくの着地時刻は？
57
6番目のしずくの生成時刻は？
78
6番目のしずくの着地時刻は？
91
7番目のしずくの生成時刻は？
15
7番目のしずくの着地時刻は？
47
8番目のしずくの生成時刻は？
1
8番目のしずくの着地時刻は？
24
9番目のしずくの生成時刻は？
74
9番目のしずくの着地時刻は？
83
4
```

<初期ソースコード>

```
#include <iostream>
using namespace std;
```



```
int main() {  
    int n;  
    int tc[100];  
    int td[100];  
    int max_count;  
  
    cout << "雨のしずくの数は何個(上限100)" << endl;  
    cin >> n;  
  
    for (int i = 0; i < n; i++) {  
        cout << i << "番目のしずくの生成時刻は?" << endl;  
        cin >> tc[i];  
        cout << i << "番目のしずくの着地時刻は?" << endl;  
        cin >> td[i];  
    }  
  
    //ここに回答を記入  
  
    cout << max_count << endl;  
    return 0;  
}
```

<回答ソースコード例>

```
#include <iostream>
using namespace std;

int main() {
    int n;
    int tc[100];
    int td[100];
    int max_count;

    cout << "雨のしずくの数は何(上限100)" << endl;
    cin >> n;

    for (int i = 0; i < n; i++) {
        cout << i << "番目のしずくの生成時刻は?" << endl;
        cin >> tc[i];
        cout << i << "番目のしずくの着地時刻は?" << endl;
        cin >> td[i];
    }
    //ここに回答を記入
    max_count = 0;
    for (int t = 0; t < 1000; t++) {
        int tmp_count = 0;
        for (int i = 0; i < n; i++) {
            if (tc[i] <= t && t <= td[i]) {
                tmp_count++;
            }
        }
        if (max_count < tmp_count) {
            max_count = tmp_count;
        }
    }
    cout << max_count << endl;
    return 0;
}
```

【2020年:概論2:第1中間:問2】不正あるじゃんけん

任意の人数でじゃんけんをする.

じゃんけんに出した手の指の数をint型の自然数で入力する.

「0」はグー (rock), 「2」はチョキ (scissors), 「5」はパー (paper),

「1, 3, 4」は不正 (illegal) とし, 「6」の入力で終了する.

不正である「1, 3, 4」の手は削除し,

正規の手である「0, 2, 5」を出した人のみで,

勝者, 敗者, 引分の人数を表示するプログラムを作成しなさい.

※ 一人でじゃんけんをしたら引き分けとする.

入力される値:じゃんけんに出した指の数

期待される出力値:不正, 勝者, 敗者, 引分の各人数

出力例(1)

```
5 4 3 2 1 0 6
不正:3, 勝者:0, 敗者:0, 引分:3
```

出力例(2)

```
5 5 2 2 1 1 1 1 5 5 6
不正:4, 勝者:2, 敗者:4, 引分:0
```

出力例(3)

```
0 0 0 0 1 3 4 6
不正:3, 勝者:0, 敗者:0, 引分:4
```

<初期ソースコード>

```
#include <iostream>
#include <vector>
#include <algorithm>
using namespace std;

int main() {
    int input, rock = 0, paper = 0, scissors = 0;
    int illegal = 0, win = 0, lose = 0, draw = 0;
    vector<int> hand;

    cin >> input;
    while (input <= 5) {
        hand.push_back(input);
        cin >> input;
    }

    cout << "不正:" << illegal
         << ", 勝者:" << win
         << ", 敗者:" << lose
         << ", 引分:" << draw;
}
```

<回答ソースコード例>

```
#include <iostream>
#include <vector>
#include <algorithm>
using namespace std;

int main() {
    int input, rock = 0, paper = 0, scissors = 0;
    int illegal = 0, win = 0, lose = 0, draw = 0;
    vector<int> hand;

    cin >> input;
    while (input != 6) {
        if (input == 1 || input == 3 || input == 4) {
            illegal++;
        }
        else {
            hand.push_back(input);
            if (input == 0) rock++;
            if (input == 2) scissors++;
            if (input == 5) paper++;
        }
        cin >> input;
    }
    if (hand.size() == 1) {
        win = 0;
        lose = 0;
        draw = 1;
    }
    else if (rock >= 1 && paper >= 1 && scissors >= 1) {
        win = 0;
        lose = 0;
        draw = hand.size();
    }
    else if (rock >= 1 && paper >= 1) {
        win = paper;
        lose = rock;
        draw = 0;
    }
    else if (paper >= 1 && scissors >= 0) {
        win = scissors;
        lose = paper;
        draw = 0;
    }
    else if (rock >= 1 && scissors >= 1) {
        win = rock;
        lose = scissors;
        draw = 0;
    }
    else{
        win = 0;
        lose = 0;
        draw = hand.size();
    }
}
```

```
}  
cout << "不正:" << illegal  
      << ", 勝者:" << win  
      << ", 敗者:" << lose  
      << ", 引分:" << draw;  
}
```