

課題部門：発表順（登録番号）

77 (88888)

タイトル：ドリフ大爆笑 2020

～もしも課題部門が・・・～

学校名：ドリフ工業高等専門学校

学生名：荒井 注

仲本工事

高木ブー

加藤 茶

志村けん

指導教員：いかりや長介

## Contents

<code>bubblesort.c</code> . . . . .	2
<code>selectionsort.c</code> . . . . .	4
<code>quicksort.c</code> . . . . .	6

**bubblesort.c**

```
1  //
2  // バブルソート
3  //
4  #include <stdio.h>
5  #include <stdlib.h>
6  #include <time.h>
7
8  #define N 30
9
10 void bubblesort(int *, int);
11
12
13 int main(int argc, char **argv)
14 {
15     int x[N]; // ソートすべきデータ
16     int i;    // ループ用
17
18
19     srandom(time(NULL));
20
21     for (i = 0 ; i < N ; i++)
22     {
23         x[i] = random() % 1000;
24     }
25
26     printf("--- 整列前 ---\n");
27     for (i = 0 ; i < N ; i++)
28     {
29         printf("%4d", x[i]);
30     }
31     printf("\n");
32
33     bubblesort(x, N);
34
35     printf("--- 整列後 ---\n");
36     for (i = 0 ; i < N ; i++)
37     {
38         printf("%4d", x[i]);
39     }
40     printf("\n");
41
42     return(0);
43 }
```

```
44
45
46  //
47  //  2つの整数変数を入れ替える
48  //
49  void swap(int *a, int *b)
50  {
51      int t;
52
53
54      t = *a;
55      *a = *b;
56      *b = t;
57  }
58
59
60
61  //
62  //  バブルソートによるソートルーチン
63  //  x[] : データ本体
64  //  n : データの個数
65  //
66  void bubblesort(int *x, int n)
67  {
68      int i, j;
69
70
71      for (j = 0 ; j < n - 1 ; j++)
72      {
73          for (i = 0 ; i < n - 1 - j ; i++)
74          {
75              if (*(x + i) > *(x + i + 1)) swap(x + i, x + i + 1);
76          }
77      }
78  }
```

**selectionsort.c**

```
1  //
2  //  直接選択ソート
3  //
4  #include <stdio.h>
5  #include <stdlib.h>
6  #include <time.h>
7
8  #define N 30
9
10 void selectionsort(int *, int);
11
12
13 int main(int argc, char **argv)
14 {
15     int x[N]; // ソートすべきデータ
16     int i;    // ループ用
17
18
19     srandom(time(NULL));
20
21     for (i = 0 ; i < N ; i++)
22     {
23         x[i]=random() % 1000;
24     }
25
26     printf("--- 整列前 ---\n");
27     for (i = 0 ; i < N ; i++)
28     {
29         printf("%4d", x[i]);
30     }
31     printf("\n");
32
33     selectionsort(x, N);
34
35     printf("--- 整列後 ---\n");
36     for (i = 0 ; i < N ; i++)
37     {
38         printf("%4d", x[i]);
39     }
40     printf("\n");
41
42     return(0);
43 }
```

```
44
45
46 //
47 // 2つの整数変数を入れ替える
48 //
49 void swap(int *a, int *b)
50 {
51     int t;
52
53
54     t = *a;
55     *a = *b;
56     *b = t;
57 }
58
59
60 //
61 // 選択ソートによるソートルーチン
62 // x[] : データ本体
63 // n : データの個数
64 //
65 void selectionsort(int *x, int n)
66 {
67     int i, j;
68     int min;
69     int idx;
70
71
72     for (i = 0 ; i < n - 1 ; i++)
73     {
74         min = *(x + i);
75         idx = i;
76         for (j = i + 1 ; j < n ; j++)
77         {
78             if (*(x + j) < min)
79             {
80                 min = *(x + j);
81                 idx = j;
82             }
83         }
84         swap(x + i, x + idx);
85     }
86 }
```

**quicksort.c**

```
1  //
2  // クイックソート
3  //
4  #include <stdio.h>
5  #include <stdlib.h>
6  #include <time.h>
7
8  #define N 30
9
10 void quicksort(int *, int, int);
11
12
13 int main(int argc, char **argv)
14 {
15     int x[N]; // ソートすべきデータ
16     int i;    // ループ用
17
18
19     srandom(time(NULL));
20
21     for (i = 0 ; i < N ; i++)
22     {
23         x[i] = random() % 1000;
24     }
25
26     printf("--- 整列前 ---\n");
27     for (i = 0 ; i < N ; i++)
28     {
29         printf("%4d", x[i]);
30     }
31     printf("\n");
32
33     quicksort(x, 0, N - 1);
34
35     printf("--- 整列後 ---\n");
36     for (i = 0 ; i < N ; i++)
37     {
38         printf("%4d", x[i]);
39     }
40     printf("\n");
41
42     return(0);
43 }
```

```
44
45
46 //
47 // 2つの整数変数を入れ替える
48 //
49 void swap(int *a, int *b)
50 {
51     int t;
52
53
54     t = *a;
55     *a = *b;
56     *b = t;
57 }
58
59
60 //
61 // クイックソートによるソートルーチン
62 // x[] : データ本体
63 // first: 先頭データのインデックス(0, ..., N-1)
64 // last : 末尾データのインデックス(0, ..., N-1)
65 //
66 void quicksort(int *x, int first, int last)
67 {
68     int m;      // ピボットの値
69     int a, b;   //
70     int idx;    //
71     int i;      // ループ用
72
73
74     idx = (last - first) / 2 + first;
75     m = *(x + idx); // x[first] ... x[last] の中間のx[m]の値を拾い出す
76
77     printf("\npivot = %3d\n", m);
78     printf("[ ");
79     for (i = first ; i <= last ; i++) printf("%3d ", x[i]);
80     printf("] --> ");
81
82     a = first;
83     b = last;
84
85     while(a <= b) // 交差するまで繰り返す
86     {
87         while ((a < last) && (*(x + a) < m )) a++; // 左から調べて
88                                                     // mより大きいデータが見つかるまでカーソルを移動する
```



```
88         while ((b > first) && (*(x + b) > m)) b--;    // 右から調べて
           mより小さいデータが見つかるまでカーソルを移動する
89         if (a > b) break;
90         swap(x + a, x + b);
91         a++;
92         b--;
93     }
94
95     printf("[ ");
96     for (i = first ; i < a ; i++) printf("%3d ", x[i]);
97     printf("]  [ ");
98     for (i = a ; i <= last ; i++) printf("%3d ", x[i]);
99     printf("]\n");
100
101     if (first < a - 1) quicksort(x, first, a - 1);
102     if (a < last) quicksort(x, a, last);
103 }
```