

アルゴリズムとデータ構造 : 第一回レ ポート

175751C 宮城孝明

平成 30 年 5 月 24 日

目 次

1 課題	2
------	---

2

1 課題

Listing 1: 課題プログラム

```
1  /*Program name: list2.c*/
2  #include <stdio.h>
3  #include <stdlib.h> //実行に必要な要素を取り入れている
4
5  struct vertex{          //ポインタが示しているデータ構造
6      int name, key        ; //ポインタ変数  name, key
7      struct vertex *next;
8  } *top;
9
10 int find_key(k)          //過去に変数に代入された数字を遡って探すkey
11     int k;
12 {
13     struct vertex *w;
14
15     w = top;
16     while(w != NULL){
17         if (k == w->key)
18             return 1;
19         w = w->next;
20     }
21     return 0;
22 }
23
24 int delete_key(k)        //変数に代入された数字に被りが出てしまった場合、過
    去の被ってkey
25     int k                ; //しまった数字を削除する
26 {
27     struct vertex *w, *prev, *tmp;
28
29     if(k == top->key){    /* top は NULL ではない*/
30         tmp = top;
31         top = top->next;
32         free(tmp);
33     }
34     else{
35         prev = top;
36         w = prev->next;
37         while(w != NULL){
38             if (k == w->key){
39                 prev->next = w->next;
40                 free(w);
41                 break;
42             }
43             prev = w;
44             w = w->next;
45         }
46     }
47     return 0;
48 }
49
```

```

50 |
51 | int main()                //主に実行している関数。ここで関数の呼び出しを行う。
52 | {
53 |     struct vertex *new;
54 |     int n, k, ans;
55 |
56 |     printf("\n");
57 |     printf(":::::::::::::リストへのデータ入力
        〇:::::::::::::〇\n");
58 |     printf("\n");
59 |
60 |     printf("各要素を入力して下さい\n");
61 |     top = NULL;
62 |     while(1){
63 |         printf("〇の値を入力しなさいname:");
64 |         scanf("%d",&n);
65 |         if(n < 0) break; /* が負ならば文を終了nwhile*/
66 |         printf("の値を入力して下さいkey:");
67 |         scanf("%d",&k);
68 |         if(find_key(k) == 1)          {//変数に代入された
            数字が過去にあったかを調べるkey
69 |             printf("key:〇〇%d〇は既登録です\n", k);
70 |             printf("key〇が〇% のものを削除しますか?
                d\n", k);
71 |             printf("YES:1/NO:0==");
72 |             scanf("%d",&ans);
73 |             if (ans == 1) delete_key(k) ;//もし
                被りがあった際に、それを消すための作業
74 |             else printf("もう一度の値から入力して下さい
                name\n");
75 |         }
76 |         else          {//もし、被りがなかった場合新しいアドレ
            スを生成するための処理
77 |             new = (struct vertex *) malloc(
                sizeof(struct vertex));
78 |             new->name = n;
79 |             new->key = k;
80 |             new->next = top;
81 |             top = new;
82 |         }
83 |     }
84 |     printf("\n");
85 |     printf("
        -----\n"
        );
86 |     printf(" 入力したデータ  ([name,key]〇の順番で");
87 |     printf("ヘッドに近い vretex〇から出力)\n");
88 |     while(top != NULL){          //過去に代入さ
        れた数字を[name,key]のくくりにして表示する]
89 |         printf("[%d,%d]",top->name, top->key);
90 |         top = top->next;
91 |     }
92 |     printf("\n");
93 | }

```

こちらが実際の実行結果です

..... リストへのデータ入力

各要素を入力して下さい
nameの値を入力しなさい:1
key の値を入力して下さい:2
nameの値を入力しなさい:3
key の値を入力して下さい:4
nameの値を入力しなさい:5
key の値を入力して下さい:6
nameの値を入力しなさい:7
key の値を入力して下さい:8
nameの値を入力しなさい:9
key の値を入力して下さい:8
key: 8 は既登録です
key が 8 のものを削除しますか?
YES: 1 / NO: 0 == 0
もう一度name の値から入力して下さい
nameの値を入力しなさい:-1

入力したデータ ([name,key] の順番でヘッドに近い vretex から出力)
[7, 8][5, 6][3, 4][1, 2]

..... リストへのデータ入力

各要素を入力して下さい
nameの値を入力しなさい:67
key の値を入力して下さい:55
nameの値を入力しなさい:45
key の値を入力して下さい:23
nameの値を入力しなさい:44
key の値を入力して下さい:22
nameの値を入力しなさい:4
key の値を入力して下さい:1
nameの値を入力しなさい:5
key の値を入力して下さい:14
nameの値を入力しなさい:-1

入力したデータ ([name,key] の順番でヘッドに近い vretex から出力)
[5, 14][4, 1][44, 22][45, 23][67, 55]

..... リストへのデータ入力

各要素を入力して下さい
nameの値を入力しなさい:98
key の値を入力して下さい:89
nameの値を入力しなさい:78
key の値を入力して下さい:67
nameの値を入力しなさい:45
key の値を入力して下さい:444
nameの値を入力しなさい:23
key の値を入力して下さい:12
nameの値を入力しなさい:98
key の値を入力して下さい:89
key: 89 は既登録です
key が 89 のものを削除しますか?
YES: 1 / NO: 0 == 0
もう一度name の値から入力して下さい
nameの値を入力しなさい:-1

入力したデータ ([name,key] の順番でヘッドに近い vretex から出力)
[23, 12][45, 444][78, 67][98, 89]

```

:..... リストへのデータ入力 :.....

各要素を入力して下さい
nameの値を入力しなさい:12
key の値を入力して下さい:23
nameの値を入力しなさい:34
key の値を入力して下さい:45
nameの値を入力しなさい:56
key の値を入力して下さい:67
nameの値を入力しなさい:78
key の値を入力して下さい:89
nameの値を入力しなさい:90
key の値を入力して下さい:1
nameの値を入力しなさい:-1

-----
入力したデータ ([name,key] の順番でヘッドに近い vretex から出力)
90, 1][78, 89][56, 67][34, 45][12, 23]

```

これらの結果より、正しく実行していることがわかる。このプログラムは、name と key にそれぞれ値を代入し、それをメモリに保存する。これを while の条件を満たしている間は、ずっと同じ入力の動作を実行してくれます。そして、name に-1 が代入されると、それまでに入力された数字を name と key のくくりにして出力してくれます。