

Tugas praktikum Algoritma dan struktur data IF-03-03

Nama : Rahmadi Rafiansyah

Nim : 1203230075

Kelas : IF 03-03


Source code :

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <stdbool.h>
4
5 typedef struct element *address;
6 struct element
7 {
8     int info;
9     address next;
10    address prev;
11 };
12
13 typedef struct
14 {
15     address first;
16     address last;
17 } list_integer;
18
19 void Createlist(list_integer *a)
20 {
21     a->first = NULL;
22     a->last = NULL;
23 }
24
25 address CreateLm(int isi)
26 {
27     address b = (address)malloc(sizeof(struct element));
28     b->info = isi;
29     b->next = NULL;
30     b->prev = NULL;
31
32     return b;
33 }
34
35 void InsertFirst(list_integer *a, address b)
36 {
37     if ((a->first == NULL) && (a->last == NULL))
38     {
39         a->first = b;
40         a->last = b;
41         b->next = a->last;
42         b->prev = a->first;
43     }
44     else
45     {
46         b->next = a->first;
47         b->prev = a->last;
48         a->first->prev = b;
49         a->last->next = b;
50         a->first = b;
51     }
52 }
53
54 void InsertLast(list_integer *a, address b)
55 {
56     if ((a->first == NULL) && (a->last == NULL))
57     {
58         a->first = b;
59         a->last = b;
60         b->next = a->last;
61         b->prev = a->first;
62     }
63     else
64     {
65         b->prev = a->last;
66         a->last->next = b;
67         a->last = b;
68         a->first->prev = a->last;
69         a->last->next = a->first;
70     }
71 }
```

```

1 void sortingAscending(list_integer *a, address new_node)
2 {
3     if (a->first == NULL)
4     {
5         a->first = new_node;
6         a->last = new_node;
7         new_node->next = new_node;
8         new_node->prev = new_node;
9     }
10    else if (a->first->info >= new_node->info)
11    {
12        address last = a->last;
13        new_node->next = a->first;
14        new_node->prev = last;
15        last->next = new_node;
16        a->first->prev = new_node;
17        a->first = new_node;
18    }
19    else
20    {
21        address curr = a->first;
22        while (curr->next != a->first && curr->next->info < new_node->info)
23        {
24            curr = curr->next;
25        }
26        new_node->next = curr->next;
27        new_node->prev = curr;
28        curr->next->prev = new_node;
29        curr->next = new_node;
30        if (curr == a->last)
31        {
32            a->last = new_node;
33        }
34    }
35 }
36
37 void Ascending(list_integer *a)
38 {
39     if (a->first == NULL || a->first == a->last)
40     {
41         return;
42     }
43
44     list_integer new_list;
45     CreateList(&new_list);
46
47     address curr = a->first;
48     address next;
49
50     do
51     {
52         next = curr->next;
53         curr->next = curr->prev = curr;
54         sortingAscending(&new_list, curr);
55         curr = next;
56     }
57     while (curr != a->first);
58
59     a->first = new_list.first;
60     a->last = new_list.last;
61 }
62
63 void TampilkanDaftar(list_integer a)
64 {
65     if (a.first == NULL)
66     {
67         printf("\n[Daftar Kosong!]\n");
68     }
69     else
70     {
71         address current = a.first;
72         do
73         {
74             printf("Di memori : %p, adalah Data ke %d\n", (void*)current, current->info);
75             current = current->next;
76         } while (current != a.first);
77         printf("\n");
78     }
79 }

```



```
1  int main()
2  {
3      list_integer a;
4      CreateList(&a);
5
6      int N, isi;
7      address b;
8
9      printf("Masukkan jumlah data : ");
10     scanf("%d", &N);
11
12     for (int i = 0; i < N; i++)
13     {
14         printf("Masukkan data ke-%d: ", i + 1);
15         scanf("%d", &isi);
16         b = CreateLm(isi);
17         InsertLast(&a, b);
18     }
19
20     printf("List data sebelum pengurutan :\n");
21     TampilkanDaftar(a);
22
23     Ascending(&a);
24
25     printf("List data setelah pengurutan :\n");
26     TampilkanDaftar(a);
27
28     return 0;
29 }
```

Penjelasan source code :

1. CreateList
Fungsi ini digunakan untuk menginisialisasi sebuah daftar kosong dengan mengatur pointer first dan last dari daftar menjadi NULL.
2. CreateLm
Fungsi ini digunakan untuk membuat sebuah elemen baru dalam daftar dengan nilai yang diberikan oleh parameter isi. Fungsi ini mengalokasikan memori untuk elemen baru, mengisi field info dengan isi, dan mengatur pointer next dan prev menjadi NULL. Fungsi ini mengembalikan alamat dari elemen yang baru dibuat.
3. InsertFirst
Fungsi ini digunakan untuk menambahkan elemen b pada awal daftar. Jika daftar masih kosong, maka elemen b menjadi elemen pertama dan terakhir. Jika tidak, elemen b ditambahkan di awal daftar dan pointer diatur untuk mempertahankan sifat circular dari list.
4. InsertLast
Fungsi ini digunakan untuk menambahkan elemen b pada akhir daftar. Jika daftar masih kosong, maka elemen b menjadi elemen pertama dan terakhir. Jika tidak, elemen b ditambahkan di akhir daftar dan pointer diatur untuk mempertahankan sifat circular dari daftar.
5. sortingAscending
Fungsi ini digunakan untuk menyisipkan elemen baru new node ke dalam daftar a dalam urutan ascending. Fungsi ini mencari posisi yang tepat untuk elemen baru dan menambahkannya sambil menjaga sifat circular dari daftar.
6. Ascending
Fungsi ini digunakan untuk mengurutkan elemen-elemen dalam daftar a dalam urutan ascending. Fungsi ini membuat daftar baru new_list dan memindahkan elemen-elemen dari daftar asli ke daftar baru dalam urutan yang terurut dengan memanggil fungsi sortingAscending.
7. TampilkanDaftar
Fungsi ini digunakan untuk menampilkan semua elemen dalam daftar a. Jika list kosong, fungsi ini mencetak pesan "[Daftar Kosong!]". Jika tidak, fungsi ini mencetak alamat memori dan nilai dari setiap elemen dalam daftar secara berurutan.
8. Main
Fungsi main merupakan fungsi utama dari program ini yang dapat membuat daftar baru a, meminta pengguna untuk memasukkan jumlah data N, memasukkan nilai N untuk menambahkannya ke dalam daftar dengan fungsi InsertLast, menampilkan daftar sebelum dan sesudah pengurutan menggunakan fungsi Ascending.