Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

Лабораторна робота №2.1
з дисципліни
«Алгоритми і структури даних»

Виконав:
студент групи ІМ-11
Шевирьов Владислав Олегович
номер у списку групи: 27

Перевірила:
Молчанова А. А.

Київ 2022

Постановка задачі

1. Створити список з ***n*** (***n***>0) елементів (***n*** вводиться з клавіатури), якщо інша кількість елементів не вказана у конкретному завданні.

2. Тип ключів (інформаційних полів) задано за варіантом.

3. Значення елементів списку взяти самостійно такими, щоб можна було продемонструвати коректність роботи алгоритму програми. Введення значень елементів списку можна виконати довільним способом (випадкові числа, формування значень за формулою, введення з файлу чи з клавіатури).

4. Вид списку (черга, стек, дек, прямий однозв'язний лінійний список, обернений однозв'язний лінійний список, двозв'язний лінійний список, однозв'язний кільцевий список, двозв'язний кільцевий список) вибрати самостійно з метою найбільш доцільного рішення поставленої за варіантом задачі.

5. Виконати над створеним списком дії, вказані за варіантом, та коректне звільнення пам'яті списку.

6. **При виконанні заданих дій, виводі значень елементів та звільненні пам'яті списку вважати, що довжина списку (кількість елементів *n* чи 2*n*) *невідома* на момент виконання цих дій.**

7. Повторювані частини алгоритму необхідно оформити у вигляді процедур або функцій (для створення, обробки, виведення та звільнення пам'яті списків) з передачею списку за допомогою параметра(ів).

Варіант 27

Варіант 27

Ключами елементів списку є цілі числа. Перекомпонувати список так, щоб спочатку розташовувались додатні, потім нульові, а за ними від'ємні елементи, не змінюючи початкового взаємного розташування елементів. Наприклад:

початковий файл: -1 0 5 -9 8 -3 5 0 -7 4

результат: 5 8 5 4 0 0 -1 -9 -3 -7.

При необхідності дозволяється використати ще один список, інші структури даних, крім простих змінних, використовувати не дозволяється.

Текст програми

```c
#include <stdio.h>
#include <stdlib.h>

struct node {
    int value;
    struct node *next;

};
struct node *newnode, *head = NULL, *temp;

void create() {
    printf("enter new n value \n");
    int n;
    scanf("%d", &n);
    for ( int i = 1; i <= n; i++ ) {
        newnode = (struct node *) malloc(sizeof(struct node));
        printf("enter new node value \n");
        scanf("%d", &newnode->value);
        newnode->next = NULL;


        if (head == NULL) {
            head = newnode;
        } else {
            temp = head;
            while (temp->next != NULL) {
                temp = temp->next;
            }
            temp->next = newnode;
        }
    }
}

void display() {
    if(head == NULL)
    {
        printf("double linked list is empty \n");

    }
```

```
    else
    {
        temp = head;
        while(temp != NULL)
        {
            printf("%d => ",temp->value);
            temp = temp->next;
        }
    }
}

void sort() {
    struct node *i, *j;
    int num;
    for(j = head; j->next != NULL; j=j->next) {
        for(i=j->next; i != NULL; i=i->next) {
            if (i->value < 0) {
                num = i->value;
                i->value = j->value;
                j->value = num;
            }
            if (i->value == 0) {
                num = i->value;
                i->value = j->value;
                j->value = num;
            }
            if (i -> value > 0) {
                num = i->value;
                i->value = j->value;
                j->value = num;
            }
        }
    }
}
void delete() {
    struct node *tmp, *next1;
    tmp = head;
    next1 = NULL;
    while (tmp) {
        next1 = tmp->next;
        free(tmp);
        tmp = next1;
    }
    (head) = NULL;
}

int main() {
    create();
    display();
    printf("sorted doubly linked list\n");
    sort();
    display();
    printf("deleted doubly linked list\n");
    delete();
    display();
    return 0;
}
```

Результати тестування програми

```
-1 => 2 => 0 => -4 => 1 => sorted doubly linked list
1 => 2 => 0 => -4 => -1 => deleted doubly linked list
double linked list is empty
```

```
0 => 1 => -9 => -3 => 0 => 1 => 2 => 5 => sorted doubly linked list
5 => 2 => 1 => 1 => 0 => 0 => -3 => -9 => deleted doubly linked list
double linked list is empty
```

```
-1 => 2 => 0 => 2 => -3 => 1 => 3 => -5 => 0 => 3 => sorted doubly linked list
3 => 3 => 1 => 2 => 2 => 0 => 0 => -5 => -3 => -1 => deleted doubly linked list
double linked list is empty
```