

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТІ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”

Кафедра систем штучного інтелекту

Лабораторна робота
з дисципліни
«Алгоритмізація та програмування»

Виконав:

студент групи КН-109

Гавришків Олексій

Викладач:

Варецький Я. Ю.

Львів – 2018 р.

Лабораторна робота №11

Тема: Інформаційні динамічні структури

Мета: Знайомство з динамічними інформаційними структурами на прикладі одно- і двонаправлених списків.

Постановка завдання

Варіант 3

Написати програму, у якій створюються динамічні структури й виконати їхню обробку у відповідності зі своїм варіантом. Записи в лінійному списку містять ключове поле типу `int`. Сформувати однонаправлений список. Знищити з нього `K` елементів, починаючи із заданого номера, додати елемент перед елементом із заданим ключем.

Код функцій:

- *Формування списку*

```
dint* make_dint( int n){
    dint *first;
    first=NULL;
    for (int i=n - 1;i>=0;i--){
        dint *p = malloc(sizeof(dint));
        p->value=i;
        p->next=first;
        first=p;
    }
    return first;
}
```

- *Друк списку*

```
dint *print_dint(dint * first){
    if (first==NULL){
        printf("The list is empty!");
        return NULL;
    }
    dint*p=first;
    while(p!=NULL)
    {
        printf("%d ",p->value);
        p=p->next;
    }
    printf("\n");
    return first;
}
```

- *Знищення списку*

```
dint *free_dint(dint*first){
    dint*p=first;
    while(p!=NULL)
    {
        p=p->next;
        free(p);
    }
    free(p);
    return NULL;
}
```

- Пошук елемента в списку

```
dint *search(dint * first,int id){
    if (first==NULL)return NULL;
    if(id < 0)return NULL;
    dint*p=first;
    int i = 0;
    while(p != NULL){
        if(id == i){
            return p;
        }
        p = p->next;
        i++;
    }
    return NULL;
}
```

- Вставлення елемента в список

```
dint *add_dint(dint * first,int value,int id){
    if(id < 0) return NULL;
    dint *new = malloc(sizeof(dint));
    new->value = value;
    if(id == 0){
        new->next = first;
        return new;
    }
    dint*pre = search(first,id - 1),*next = search(first,id);
    if(pre ==NULL || next == NULL)return NULL;
    pre->next = new;
    new->next = next;
    return first;
}
```

- Видалення елемента з списку

```
dint *del_dint(dint * first,int id){
    if(id < 0) return NULL;
    if(id == 0){
        dint *f = first->next;
        free(first);
        return f;
    }
    dint*pre = search(first,id - 1),*el = search(first,id),*next =
search(first,id+1);
    if(pre == NULL || next == NULL || el == NULL)return NULL;
    pre->next = next;
    free(el);
    return first;
}
```

- Видалення K-елементів з списку

```
dint *del_elements(dint*first,int id,int k){
    dint *p = first;
    for(int i = 0; i < k;i++){
        p = del_dint(first,id + i);
    }
    return p;
}
```

- Запис списку в файл

```
int write_to_file(dint *first){
    FILE *f;
    if (first==NULL) return -2;
    if ((f=fopen("file11.txt", "wb"))==NULL) return -1;
    dint*p=first;
    while(p!=NULL)
    {
        fprintf(f, "%d\n", p->value);
        p=p->next;
    }
    fclose(f);
    return 1;
}
```

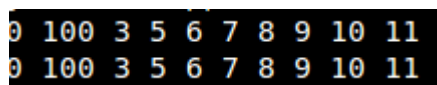
- Вивернення списку

```
dint *overturn(dint* first){
    dint*p=first;
    dint*prev = NULL;
    dint*temp;
    while(p != NULL){
        temp = p->next;
        p->next = prev;
        prev = p;
        p = temp;
    }
    return prev;
}
```

- Відновлення списку з файлу

```
dint* read_file(){
    FILE *f;
    if ((f=fopen("file11.txt", "rb"))==NULL) return NULL;
    dint *first = NULL;
    char s[15];
    char *end;
    fgets(s,12,f);
    do {
        dint *p = malloc(sizeof(dint));
        p->value = strtol(s, &end, 10);
        p->next=first;
        first=p;
        fgets(s,12,f);
    }while(!feof(f));
    fclose(f);
    return overturn(first);
}
```

Результат виконання програми:



```
0 100 3 5 6 7 8 9 10 11
0 100 3 5 6 7 8 9 10 11
```

Висновок: На цій лабораторній роботі я навчився працювати з одно- і двонаправленими списками та записувати їх у файли з можливістю відновити їх.