МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТІ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”

Кафедра систем штучного інтелекту

Звіт

про виконання лабораторної роботи №11
з дисципліни **«Алгоритмізація та програмування, частина 1»**
Варіант №2

Виконав:
студент групи КН-109
Ханас Михайло-Юрій
Викладач:
Варецький Я.Ю.

Львів – 2018 р.

Тема: "Інформаційні динамічні структури"

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

Постановка завдання:

1. Створення списку.
2. Додати елемент перед елементом із заданим ключем.
3. Знищити з нього елемент із заданим ключем.
4. Друк списку.
5. Запис списку у файл.
6. Знищення списку.
7. Відновлення списку з файлу.

Програма розв'язку завдання :

```
#include <stdio.h>
#include <stdlib.h>
typedef struct list
{
   int value;
   struct list *next;
} list_t;

list_t *print(struct list*first)
{
   if (first==NULL)
   {
      printf("The list is empty. ");
      return NULL;
   }
   struct list*p=first;
   int i=1;
   while(p!=NULL)
   {
   printf("key[%d]=%d\n",i,p->value);
   p=p->next;
   i++;
   }
   return first;
}


list_t* fill_list(int n)
{
```

```
    struct list *first = NULL;
    for (int i=n;i>0;i--)
    {
        struct list *p=malloc(sizeof(list_t));
        p->value=i;
        p->next=first;
        first=p;
    }
return first;
}

list_t *freed(struct list*first){
    struct list* p=first;
    struct list* temp;
    while(p->next!=NULL)
    {
        p=p->next;
        temp=p;
        p=temp->next;
        free(temp);

    }
    free(first);
    return NULL;
}
list_t *search(struct list * first,int id){
    if (first==NULL)return NULL;
    if(id < 0)return NULL;
    struct list*p=first;
    int i = 0;
    while(p != NULL){
        if(id == i){
            return p;
        }
        p = p->next;
        i++;
    }
    return NULL;
}
list_t *add(struct list * first,int value,int key){
    if(key < 0) return NULL;
    struct list *new = malloc(sizeof(list_t));
    new->value = value;
    if(key == 0){
```

```
        new->next = first;
        return new;
    }
    struct list*previous= search(first,key - 2);
    struct list*next = search(first,key-1);
    if(previous ==NULL || next == NULL)return NULL;
    previous->next = new;
    new->next = next;
    return first;
}
list_t *del(struct list *first,int key){
    if(key < 0) return NULL;
    if(key == 0){
        struct list *f = first->next;
        free(first);
        return f;
    }
    struct list*previous = search(first,key - 2),*el = search(first,key-1),*next =
            search(first,key);
    if(previous == NULL && next == NULL && el == NULL)return NULL;
    previous->next = next;
    free(el);
    return first;
}

list_t *overturn(struct list* first){
    struct list*p=first;
    struct list*previous = NULL;
    struct list*temp;
    while(p != NULL){
        temp = p->next;
        p->next = previous;
        previous = p;
        p = temp;
    }
    return previous;
}

int write_to_file(struct list *first){
    FILE *f;
    if (first==NULL)return -2;
    if ((f=fopen("f.txt", "w+"))==NULL) return -1;
    struct list*p=first;
    while(p!=NULL)
```

```
    {
        fwrite(&p->value, sizeof(list_t), 1, f);
        p=p->next;
    }
    fclose(f);
    return 1;
}

list_t *read_file(){
    FILE *f;
    if ((f=fopen("f.txt", "r+"))==NULL) return NULL;
    struct list *first = NULL;
    while(1){
        struct list *p = malloc(sizeof(list_t));
        fread(&p->value, sizeof(list_t), 1,f );
        p->next=first;
        first=p;
        if(feof(f))break;
    }
    fclose(f);
    return overturn(first);
}

int main()
{
    int n,value,key,dele;
    printf("Put the number of elements: ");
    scanf("%d",&n);
    struct list *arr=fill_list(n);
    print(arr);
    printf("Add element: ");
    scanf("%d",&key);
    printf("Value: ");
    scanf("%d",&value);
    if((arr = add(arr,value,key)) == NULL){
        printf("Error! Unable to add element!");
        exit(1);
    }

    printf("\n");

    print(arr);
    printf("\n");
    printf("Delete element: ");
```

```
    scanf("%d",&dele);
    if((arr = del(arr,dele)) == NULL){
        printf("Error!2");
        exit(2);
    }
    print(arr);printf("\n");
    if(write_to_file(arr) != 1){
        printf("Error!3");
        exit(3);
    }
    arr = freed(arr);
    print(arr);printf("\n");
    if((arr = read_file()) == NULL){
        printf("Error!4");
        exit(4);
    }
    print(arr);
    printf("\n");

return 0;
}
```

```
Put the number of elements: 5
key[1]=1
key[2]=2
key[3]=3
key[4]=4
key[5]=5
Add element: 3
Value: 20

key[1]=1
key[2]=2
key[3]=20
key[4]=3
key[5]=4
key[6]=5

Delete element: 4
key[1]=1
key[2]=2
key[3]=20
key[4]=4
key[5]=5
The list is empty.
key[1]=5
key[2]=4
key[3]=20
key[4]=2
key[5]=1
```

Висновок:

Під час виконання цієї лабораторної я ознайомився з динамічними інформаційними структурами на прикладі одно- і двонаправлених списків.