МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТІ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”

Кафедра систем штучного інтелекту

Лабораторна робота 11

з дисципліни

«Алгоритмізації та програмування»

Виконав:
студент групи КН-108
Левицький Богдан

Львів – 2018 р.

Варіант 15

1) Написати програму, у якій створюються динамічні структури й виконати їхню обробку у відповідності зі своїм варіантом

Записи в лінійному списку містять ключове поле типу *char (рядок символів). Сформувати двонаправлений список. Знищити К елементів з кінця списку. Додати елемент після елемента із заданим ключем.

2)

```
#include <stdio.h>
#include <string.h>

typedef struct Node
{
    int key;
    char* data;
    struct Node* next;
    struct Node* prev;

}Node;

Node* createList(char* data) // creates list
{
    Node* head = (Node*) malloc(sizeof(Node));  // C
    head->next = NULL;  // set next pointer to NULL, because list consists of 1 element
    head->prev = NULL;      // set previous pointer to NULL, because list consists of 1 element
    head->data = data;  // data to list
```

```
    head->key = 1;      // key of first element = 1
    return head;        // return pointer to first element
}

Node* putElement(Node* head,char* data,int key)  // Puts element in list in key position
{
    if(head == NULL) // creates list if it doesnt exist
    {
            Node* head = (Node*) malloc(sizeof(Node));
            head->next = NULL;
            head->prev = NULL;
            head->data = data;
            head->key = 1;
            return head;
    }

    // add element in key posititon

    Node* tmp = (Node*) malloc(sizeof(Node));        // memory for new element
    while(head->next!= NULL)                         // looking for element after which we need to put new element
    {
            if(key == head->key)
            {
                    break;
            }
            head = head->next;
```

```
    }

    // manipulate with pointer to put our new element

    tmp->prev = head;
    tmp->next = head->next;
    head->next = tmp;
    tmp->key = head->key + 1;
    tmp->data = data;

    // changing keys of next elements

    while(tmp->next != NULL)
    {
            tmp->next->key = tmp->key + 1;
            tmp = tmp->next;
    }
}

void killLastsElems(Node*head,int lasts)   // kill all element after one which we set
{
    Node* temp = head;
    int counter = 0;
    while(head != NULL && counter < lasts)
    {
            head = head->next;
            counter++;
    }
```

```
    head->prev->next = NULL;
    while(head!= NULL)
    {
            temp = head;
            head = head->next;
            free(temp);
    }
    head = NULL;
    temp = NULL;

}

void print(Node* head)   // Print the list, if it is empty print that it is empty
{
    if(head == NULL)
    {
            printf("The list is empty\n");
    }
    else
    {
            while(head != NULL)
            {
                    printf("%d\t",head->key);
                    int k = strlen(head->data);
                    for(int i = 0;i<k;i++)
                    {
                            printf("%c",head->data[i]);
                    }
```

```
                printf("\n");
                head = head->next;
            }
    }
    printf("--------------------------\n");
}

Node* freeList(Node* head)          // kill all the list
{
    Node* tmp = head;
    while(head != NULL)
    {
            tmp = head;
            head = head->next;
            free(tmp);
    }
    if(head == NULL)
    {
            printf("Successful, the list was killed\n--------------------------\n");
            return head;
    }
    else
    {
            printf("Error\n");
            exit(-1);
    }
}
```

```
void writeInFile(Node* head,char* fName)    // Writes the list in file, fname - name of the file, head- pointer to first element
{
    FILE* fp = fopen(fName,"wb");
    if(fp == NULL)
            exit(-3);
    int i =0;
    while(head != NULL && !feof(fp))
    {
            fprintf(fp,"%s ",head->data);
            head = head->next;
    }
    fclose(fp);
}

Node* listFromFile(char* fName)    // creating list from the File
{
    Node* head = NULL;
    FILE* fp = fopen(fName,"r");
    if(fp == NULL)
            exit(-4);
    char buffer[255];
    fread(buffer,sizeof(Node),1,fp);
    int key = 0;
    int i = 0;
    int count = 0;
    while(buffer[i] != '\0')
    {
```

```
            while(buffer[i] != ' ')
            {
                    i++;
            }
            char* tempbuff = malloc(sizeof(char) * (i - count) );
            int j = 0;
            while( count < i)
            {
                    tempbuff[j] = buffer[count];
                    j++;
                    count++;
            }
            tempbuff[j] = '\0';
            key++;
            i++;
            count++;
            if(head == NULL)
                    head = createList(tempbuff);
            else
                    putElement(head,tempbuff,key);
    }
    return head;
}

int main(int argc, char* argv[])
{
    char* greeting = "Hello";
    Node* head = createList(greeting);
```

```
    putElement(head,"Laboratory11",2);
    putElement(head,"done",3);
    putElement(head,"by",4);
    putElement(head,"Bohdan",6);
    putElement(head,"student",5);
    putElement(head,"Levytskyi",7);
    printf("List after adding elements\n");
    print(head);
    killLastsElems(head,4);
    printf("List after killint last 4 elements\n");
    print(head);
    writeInFile(head,"Test.txt");
    head = freeList(head);
    print(head);
    head = listFromFile("Test.txt");
   print(head);
    system("pause");
    return 0;
}
```

3)

E:\Study\Programming\lab11.exe

```
List after adding elements
1       Hello
2       Laboratory11
3       done
4       by
5       Bohdan
6       student
7       Levytskyi
---------------------------
List after killinя last elements after 4th
1       Hello
2       Laboratory11
3       done
4       by
---------------------------
Successful, the list was killed
---------------------------
The list is empty
---------------------------
1       Hello
2       Laboratory11
3       done
4       by
---------------------------
Для продолжения нажмите любую клавишу . . .
```

Test — Блокнот

Файл Правка Формат Вид Справка

Hello Laboratory11 done by