

# Лабораторна робота №22.

## Загальне завдання

Поширити попередню лабораторну роботу таким чином:

- **Ви**
- **Р**
  - при введенні інформації про “базовий тип” (немає різниці, чи з клавіатури, чи з файлу), організувати перевірку відповідності таким критеріям з використанням регулярних виразів:
    - – можна вводити тільки кириличні символи, латинські символи, цифри, пропуски, розділові знаки;
    - – не повинно бути пропусків та розділових знаків, які повторюються;
    - – перше слово не повинно починатися з маленького символу;
- **С**
  - у клас-список додати метод, що виводить на екран список усіх об’єктів, які мають одне або більше полів з щонайменше двома словами (перевірку організувати за допомогою регулярних виразів).

## • Опис роботи

Програма працює шляхом введення даних від користувача з файлу і їх подальшого аналізу з виведенням на екран та в файл результату. Також в кінці відбувається тест. Але якщо використовується `debug`, то також буде виведена інформація про час роботи програми, дату, а також ім'я функції. Також програма має в собі регулярні вирази, які використовуються для перевірки критерієм згідно завданням

## • Функціональне призначення

Дана програма працює за допомогою використання динамічної пам'яті, а точніше маніпуляцією динамічної пам'яті, також використовується файл. Для макросу використовується проста "функція" в якості макросу. А в якості перевірки заданим критерієм використовуються регулярні вирази.

## • Опис логічної структури

Дана програма може бути використана для створення динамічного списку, двонаправленого, який буде зберігати собі дані і буде легкий для підправлення (видалення і введення даних). Також ця програма буде виводити ім'я функції і дату за певних умов і використовувати для перевірки критерієм регулярні вирази.

Функція main містить в собі виклик функцій, а також корисних властивостей для функцій

Функція append збільшує розмір списку

Функція push вставляє в кінець елемент

Функція insertAfter вставляє елемент після певного індексу

Функція printList по

```
regex_t regex;
regcomp(&regex, "[a-z]+[0-9]+");
// eng + rus + nums

int return_value =
regcomp(&regex, "^[a-z]+[0-9]+$");
// upper check

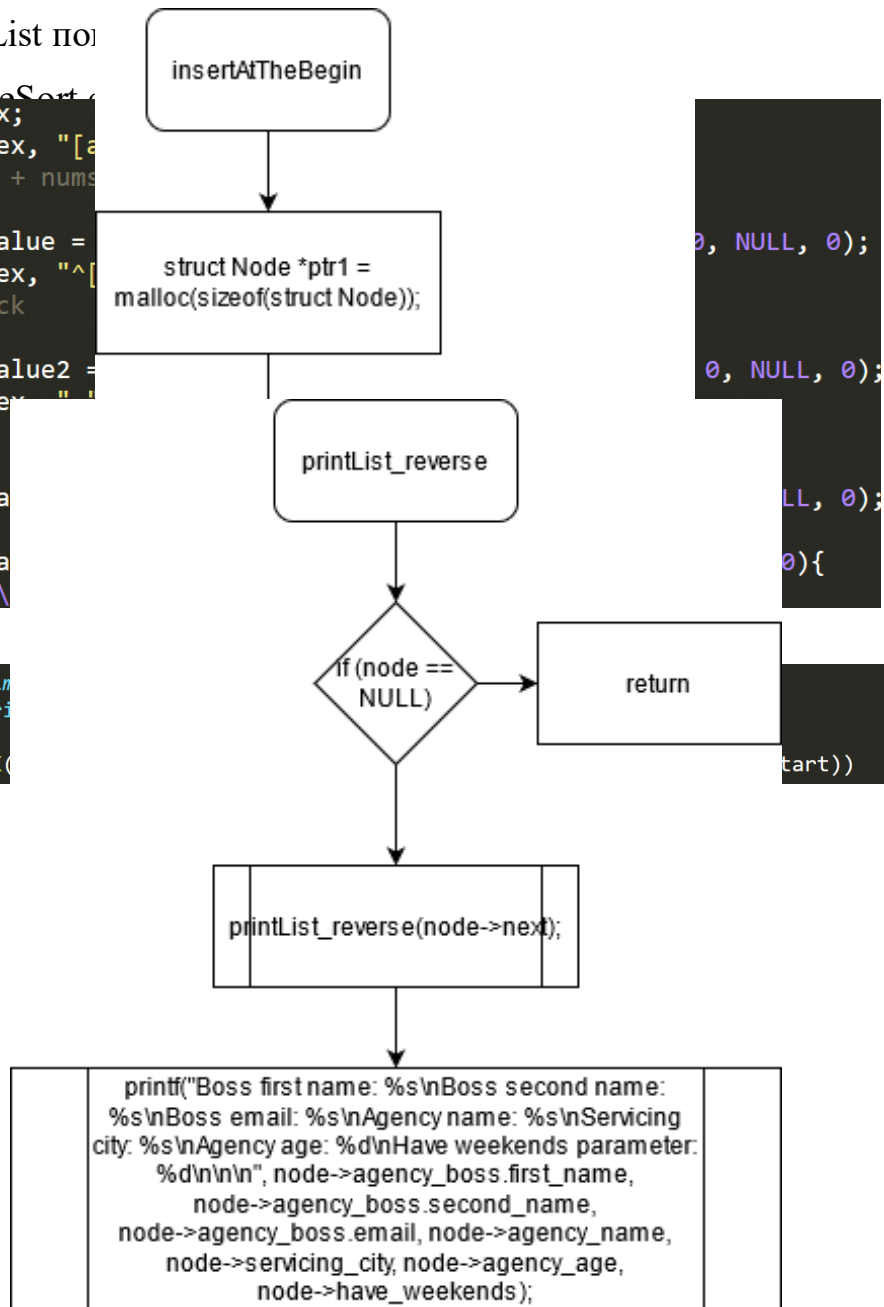
int return_value2 =
regcomp(&regex, "[a-z]+[0-9]+$");
// 2+ words

int return_va

if (return_va
printf("\n
```

```
#define TASK()  tim
pri

#define TIME_CHECK(
```



```
0, NULL, 0);

0, NULL, 0);

LL, 0);

0){
```

```
start))
```

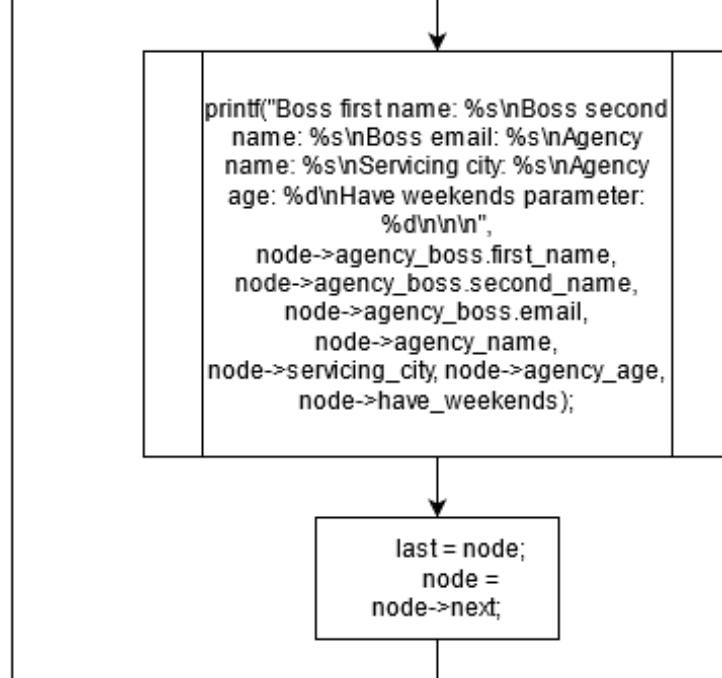


рис. 4 - printList

- **Важливі елементи програми**

```
struct Agency* new_node = malloc(sizeof(struct Agency));
```

```
struct Agency* last = *head_ref;
```

```
strcpy(new_node->agency_boss.first_name, "x");
```

```
strcpy(new_node->agency_boss.second_name, "x");
```

```
strcpy(new_node->agency_boss.email, "x");
```

```
strcpy(new_node->agency_name, "x");
```

```
strcpy(new_node->servicing_city, "x");
```

```
new_node->agency_age = 5;
```

```
new_node->have_weekends = 0;
```

```
new_node->next = NULL;
```

```
if (*head_ref == NULL) {
```

```
    new_node->prev = NULL;
```

```
    *head_ref = new_node;
```

```

        return;
    }

    while (last->next != NULL)

        last = last->next;

    last->next = new_node;

    new_node->prev = last;

```

```

    struct Node *temp = start;
    printf("\n");
    while (temp!=NULL)
    {
        printf("%d ", temp->data);
        temp = temp->next;
    }

```

```

#define TASK()  time_t time1 = time(NULL);\
                printf("Function %s. Time %s ", __func__, ctime(&time1))

#define TIME_CHECK(start,end) printf("Program worked for %d seconds\n", (end)-(start))

```

```

    struct Node *ptr1 = malloc(sizeof(struct Node));

```

```
ptr1->data = data;

ptr1->next = *start_ref;

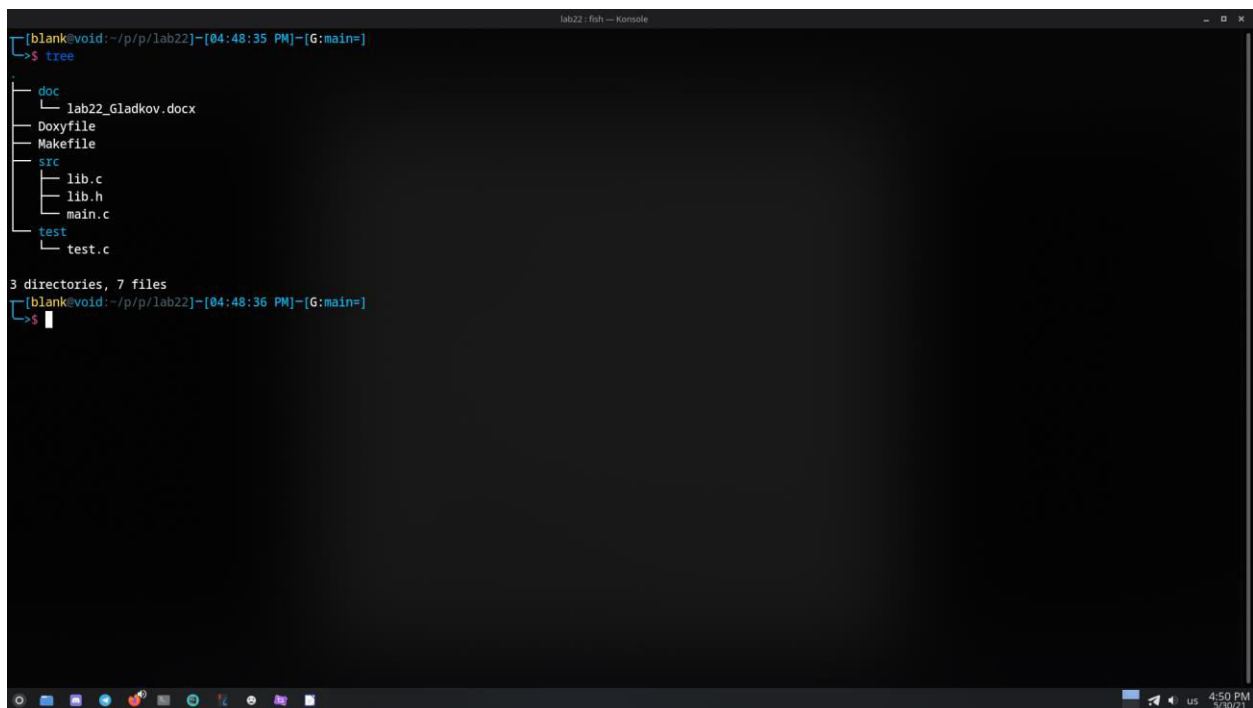
if (*start_ref != NULL) {

    (*start_ref)->prev = ptr1;

}

*start_ref = ptr1;
```

- **Структура**



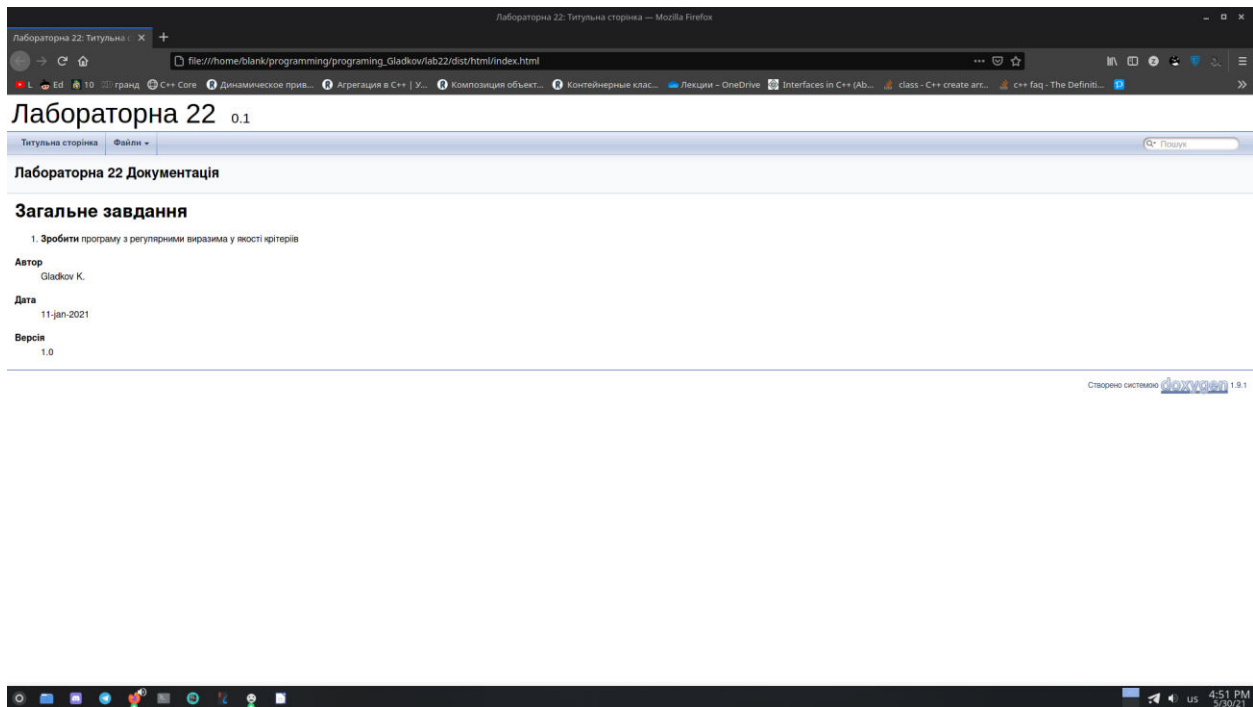
The screenshot shows a terminal window titled 'lab22 : fish — Konsole'. The prompt is '[blank@void: ~/p/p/lab22]~[04:48:35 PM]~[G:main=]'. The user has entered the command 'tree', and the output is as follows:

```
tree
.
├── doc
│   └── lab22_Gladkov.docx
├── Doxyfile
├── Makefile
├── src
│   ├── lib.c
│   ├── lib.h
│   └── main.c
├── test
└── test.c

3 directories, 7 files
```

The prompt is now '[blank@void: ~/p/p/lab22]~[04:48:36 PM]~[G:main=]' and the cursor is at the end of the line.

- **Doxygen**



- **Варіанти використання**

Дана програма може бути використана для роботи з динамічними списками. Також ця програма буде виводити ім'я функції і дату за певних умов. І вона прозводить перевірку критерієм.

## **Висновки**

У даній лабораторній роботі був придбаний досвід роботи з динамічними списками і роботою з макросами. Також був придбаний досвід роботи з регулярними виразом і складанням з їх допомогою критерій, за яким працює програма.