

### **Завдання на оцінку “задовільно” та “добре”:**

Для попередньо розробленого функціоналу по роботі з прикладною областю за допомогою макровизначень зробити наступну “кастомізацію”: якщо визначене макровизначення `DEBUG`, то при виклику кожної функції роботи з прикладною областю виконувати таке:

- виводити поточну дату та час;
  - виводити ім'я поточної функції;
  - при завершенні програми вивести загальний час роботи програми.
- **Ви**
  - **Р**
  - **Завдання на оцінку “відмінно”:**
  - виконати основне завдання;
  - змінити модульні тести таким чином, щоб:
    - при запуску кожного тесту виводити ім'я тесту, а по завершенні тесту виводити час, що було витрачено на виконання даного тесту;
    - по завершенні всіх тестів було виведено час їх виконання.
  - **С**

### **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 інформацію про час роботи програми, дату, а також ім'я функції.

### **Функціональне призначення**

Дана програма працює за допомогою використання динамічної пам'яті, а точніше маніпуляцією динаміч. пам'яті, також використовується файли. Для макросу використовується проста "функція" в якості макросу

### **Опис логічної структури**

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

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

Функція append збільшує розмір списку

Функція push вставляє елемент в кінець списку

Функція insertAfter вставляє елемент після певного вузла

Функція printList повертає список

Функція bubbleSort сортує список

Функція swap служить для обміну значеннями

Функція test\_append перевіряє роботу функції append

Функція print\_reverse

Функція insertAtTheBegin

Функція printList

```
#define TASK() time_t start = time(NULL);
#define TIME_CHECK() printf("Time taken: %d\n", (int)(time(NULL) - start))
```

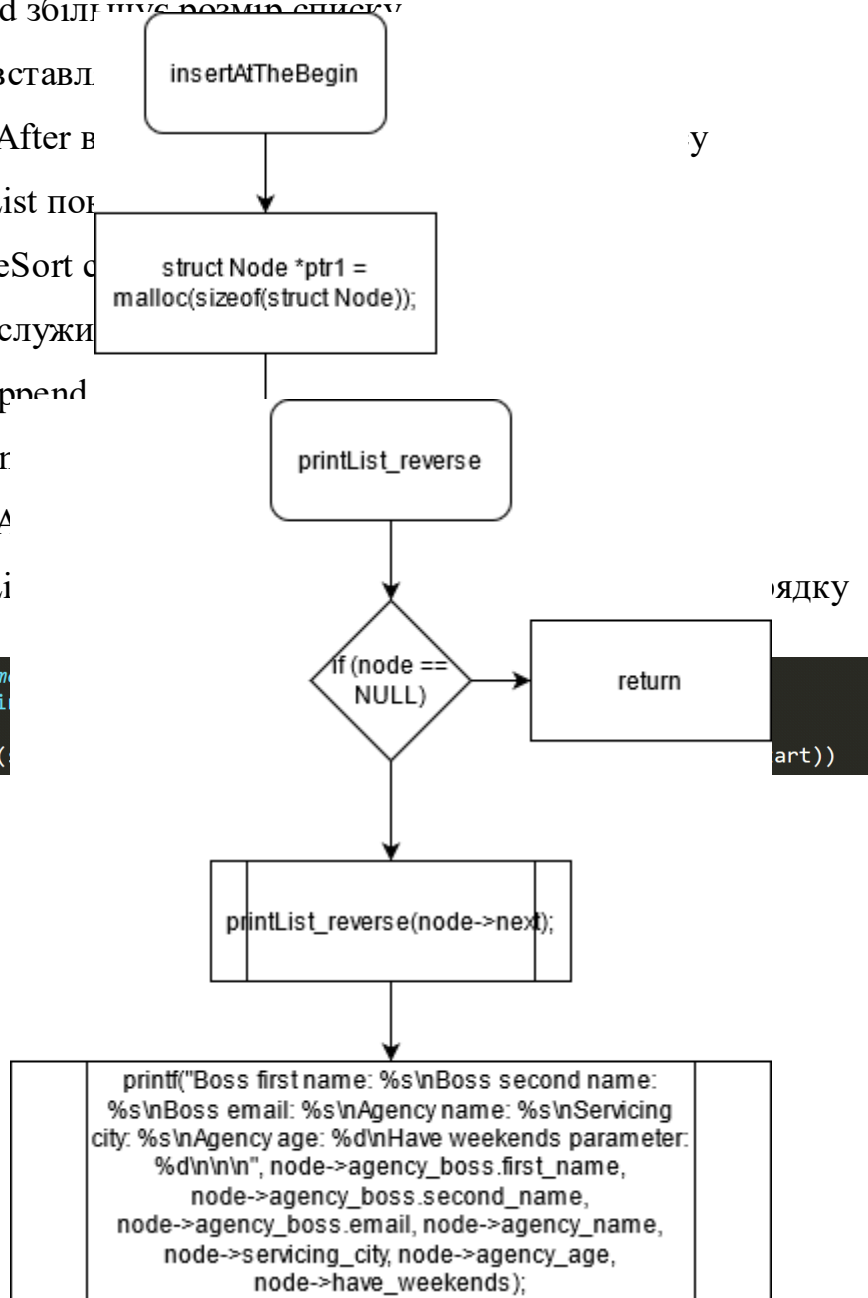


рис. 2 - printList\_reverse

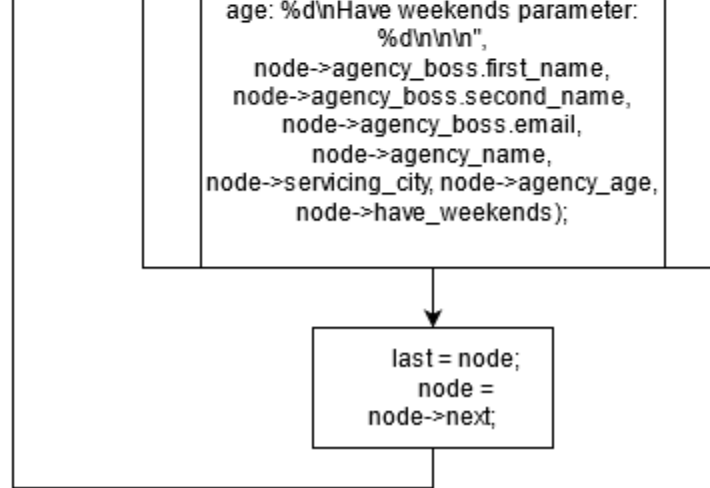


рис. 4 - printList

- ## 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;
```

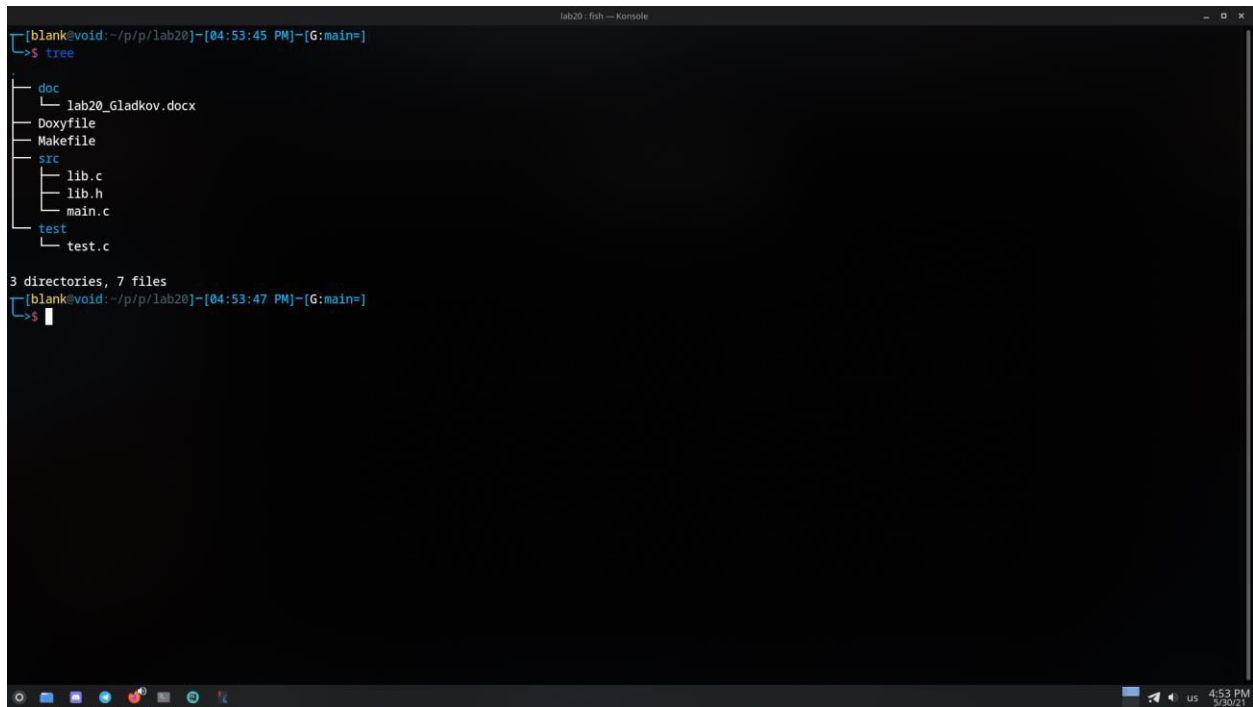
```
ptr1->next = *start_ref;
```

```
if (*start_ref != NULL){
```

```
    (*start_ref)->prev = ptr1;
```

```
}  
  
*start_ref = ptr1;
```

- **Структура**

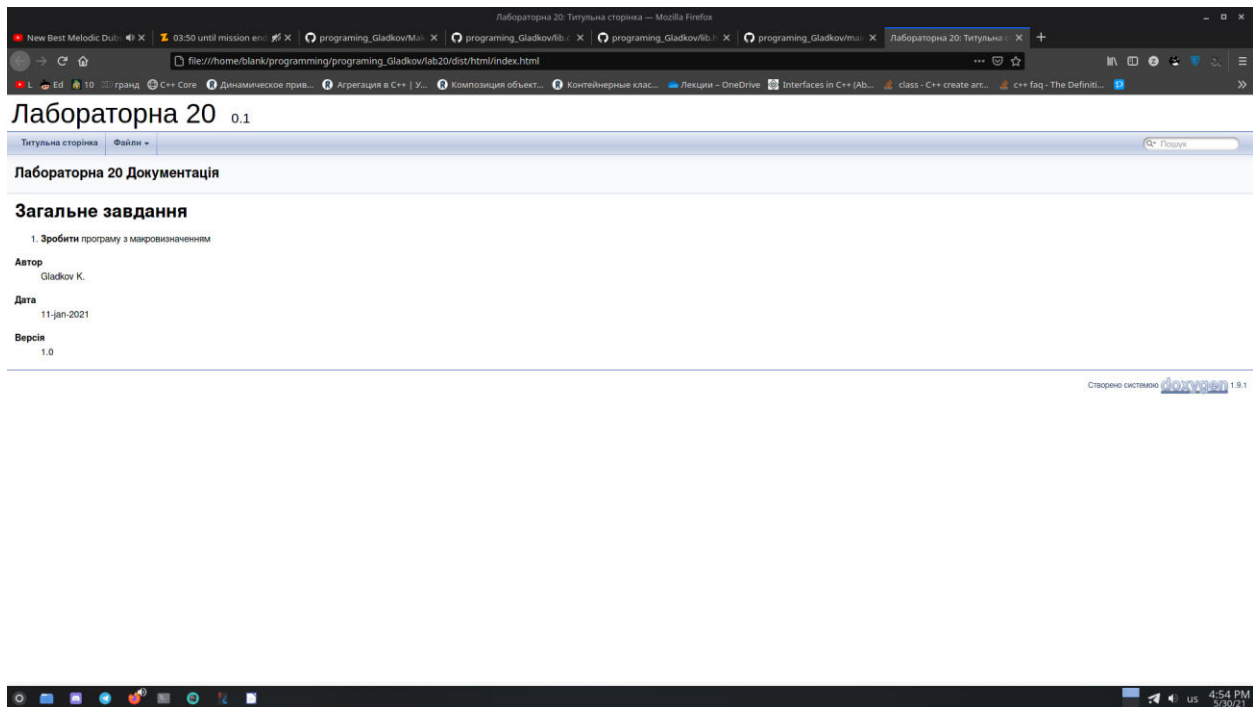


The screenshot shows a Windows console window titled "lab20 : fish — Konsole". The prompt is "[blank@void: ~/p/p/lab20]~[04:53:45 PM]~[G:main=]". The user has entered the command `tree`, and the output is a directory tree structure:

```
.  
├── doc  
│   └── lab20_Gladkov.docx  
├── Doxyfile  
├── Makefile  
├── src  
│   ├── lib.c  
│   ├── lib.h  
│   └── main.c  
└── test  
    └── test.c
```

Below the tree, it says "3 directories, 7 files". The prompt then changes to "[blank@void: ~/p/p/lab20]~[04:53:47 PM]~[G:main=]". The user has entered a dollar sign `$` 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.

## **Висновки**

У даній лабораторній роботі був придобаний досвід роботи з динамічними списками і роботою з макросами