

Звіт

До лабораторної роботи №3

Виконав:

Студент Руденко Альберт ПЗ-16

Тема. Зміст та роль тестування у життєвому циклі розробки програмного забезпечення.

Мета роботи. Вивчити основні методи та принципи тестування на прикладі створення власної програми.

Теоретичні відомості

9.Перевірка логічної схеми програми полягає у аналіз логічних розгалужень і виявлення логічних помилок.

27.Тестування класифікують : димове тестування, санітарне, регресійне тестування, альфа- та бета-тестування.

39. Статичний аналіз коду – це уважна перевірка коду (найчастіше – вихідного коду програми) з метою виявлення в ній семантичних помилок, без виконання програми. Код читається та аналізується спеціалістами. Відрізняється від динамічного відсутністю виконання на реальному чи віртуальному процесорі.

Постановка завдання

1. Провести функціональне тестування програми для роботи з динамічними списками (розробленої у ході виконання лабораторної роботи з дисципліни «Основи програмування»), використавши такі тести:

1. димовий тест,
2. тест нормальних умов,
3. тест екстремальних(граничних) умов,
4. тест виняткових умов.

1.1. Результати тестів 1, 2, 3 та 4 оформити у вигляді таблиці (за зразок взяти таблицю 2). У таблиці інформація заноситься в зрозумілій формі, достатньо повно, щоб не вимагалось додаткових (усних) пояснень. Для представлення *тестових даних* та *фактичних результатів* бажано використовувати посилання на зображення екранів чи відповідні файли. В *очікуваних результатах* зафіксувати, яка функція була протестована та відповідно, що є очікуваним результатом. Якщо помилки не знайдено у *Тип звіту* ставимо знак -. Примітки використовуємо для додаткових пояснень.

1.2. Знайдені дефекти виправити. Провести повторне тестування, результати якого оформити окремою таблицею.

2. Окремо провести тестування логічної схеми програми (головна програма та дві функції-підпрограми). Для цього зобразити її у вигляді графів управління. Вершини графів пронумерувати. Записати усі отримані маршрути роботи програми, використовуючи номери вершин.

2.1. Для кожного маршруту провести тестування і записати результати тестування у звітній таблиці.

2.2. Знайдені дефекти виправити. Провести повторне тестування, результати якого оформити окремою таблицею.

1. Функціональне тестування

№ з/п	Тестові дані	Фактичні результати	Очікуванні Результати	Ступінь критичності	Тип звіту	П р и м і т к и
Тест нормальних умов						
1	літера – а, що запускає введення даних test Test 11.11.1111 1 1 1	Коректне зчитування і відображення(рис1)	Додавання стрічки до списку	-	-	-
2	Вхідний файл : student_list.txt	Коректне читання файлу та відображення інформації	Правильне зчитування та виведення	-	-	-
Тест граничних умов						
1	Запуск з пустим файлом	Проблема у створені списку та виведення некоректних даних Рис2	Додавання до пустого списку	Критичний	Помилка кодування	-
2	Запуск з 255 символами	Нормальне додавання до списку	Вивід помилки	-	-	-
Тест виняткових ситуацій						
1	Запуск програми без вхідного файлу	Проблема з читанням пам'яті Рис3	Запуск діалогу про додавання до списку елемента	Критичний	Помилка кодування	-

```
delete or add?a
Enter data: test|test|11.11.1111|3|3|3

Student1 student1 05.06.2003 5 5 5
Student2 student2 07.06.2003 3 3 3
Student3 student3 08.06.2003 2 3 2
Student4 student4 06.06.2003 4 4 4
test test 11.11.1111 3 3 3
```

Рис1

```

delete or add?a
Enter data: o o o 6 6 6

(null) (null) (null) 0 0 0
o o o 6 6 6
o o o 6 6 6

```

Рис2

```

Starting program: /home/a.out

Breakpoint 1, main () at main.c:99
99      while (getc(student_list) != EOF) {
(gdb) next

Program received signal SIGSEGV, Segmentation fault.
0x00007ffff7e565ad in _IO_getc (fp=0x0) at getc.c:37
37      getc.c: No such file or directory.
(gdb)

```

Рис3

```

delete or add?a
Enter data: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis partu
rient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. Donec pede justo, fringill
a vel, aliquet nec, vulputate eget, arcu. In enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo. Nullam dictum felis eu pede mollis pretium. Integer tincidun
t. Cras dapibus. Vivamus elementum semper nisi. Aenean vulputate eleifend tellus. Aenean leo ligula, porttitor eu, consequat vitae, eleifend ac, enim. Aliquam lorem a
hte, dapibus in, viverra quis, fegiat a, tellus. Phasellus viverra nulla ut metus varius laoreet. Quisque rutrum. Aenean imperdiet. Etiam ultricies nisi vel augue. C
urabitur ullamcorper ultricies nisi. Nam eget dui. Etiam rhoncus. Maecenas tempus, tellus eget condimentum rhoncus, sem quam semper libero, sit amet adipiscing sem ne
que sed ipsum. Nam quam nunc, blandit vel, luctus pulvinar, hendrerit id, lorem. Maecenas nec odio et ante tincidunt tempus. Donec vitae sapien ut libero venenatis fa
ucibus. Nullam quis ante. Etiam sit amet orci eget eros faucibus tincidunt. Duis leo. Sed fringilla mauris sit amet nibh. Donec sodales sagittis magna. Sed consequat,
leo eget bibendum sodales, augue velit cursus nunc,

```

Рис4

Виправлення 1 етап

```
if(student_list != NULL){  
    while (getc(student_list) != EOF) {  
        numberOfchars++;  
        fseek(student_list, 0, SEEK_SET);  
  
        char currChar = '\0';  
        for(int i = 0; (currChar = getc(student_list)) != EOF; i++) {  
            amountOfchars[i] = currChar;  
        }  
    }  
}
```

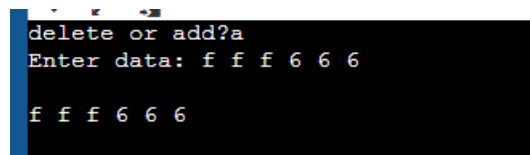
Додана перевірка чи існує файл

```
if(student_list != NULL){  
    prev->nextStudent = new_student;  
    prev = new_student;  
    print_students(head);  
}else{  
    print_students(new_student);  
}
```

Додана перевірка чи існує список

Повторне тестування

№ з/п	Тестові дані	Фактичні результати	Очікуванні Результати	Ступінь критичності	Тип звіту	П р и м і т к и
Тест нормальних умов						
1	літера – а, що запускає введення даних test Test 11.11.1111 1 1 1	Коректне зчитування і відображення	Додавання стрічки до списку	-	-	-
2	Вхідний файл : student_list.txt	Коректне читання файлу та відображення інформації	Правильне зчитування та виведення	-	-	-
Тест граничних умов						
1	Запуск з пустим файлом	Нормальне додавання до списку Рис1	Додавання до пустого списку	-	-	-
2	Зупуск з 255 символами	Нормальне додавання до списку	Вивід помилки	-	-	-
Тест виняткових ситуацій						
1	Запуск програми без вхідного файлу	Нормальне додавання до пустого списку Рис2	Запуск діалогу про додавання до списку елемента	-	-	-



```

delete or add?a
Enter data: f f f 6 6 6

f f f 6 6 6

```

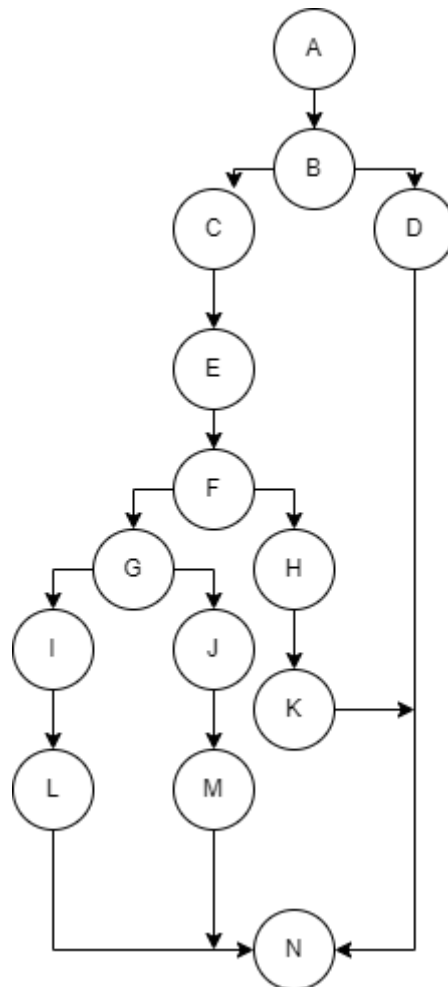
Рис1

```
delete or add?a
Enter data: f f f 5 5 5
f f f 5 5 5
```

Рис2

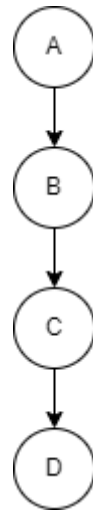
Структурне тестування

Головна функція



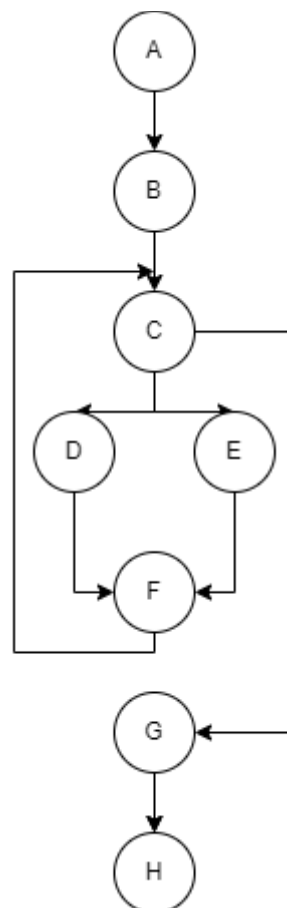
- 1)A->B->D->N
- 2)A->B->C-E-F->G->I-L->N
- 3)A->B->C-E-F->G->J-M->N
- 2)A->B->C-E-F->H-K-N

Функція make_list



1)A-B-C-D

Функція reconstruct



1)A-B-C->D-F-C->G-H

2) A-B-C->E-F-C->G-H

№ з/п	Тестові дані	Фактичні результати	Очікуванні Результати	Ступінь критичності	Тип звіту	Примітки
Main						
1	Запуск програми без файлу	Нормальне додавання до списку	Нормальне додавання до списку	-	-	-
2	Пустий файл	Нормальне додавання до списку	Нормальне додавання до списку	-	-	-
3	Додавання до списку	Нормальне додавання до списку	Нормальне додавання до списку	-	-	-
4	Видалення елементів зі списку	Нормальне видалення зі списку	Нормальне видалення зі списку	-	-	-
Функція make_list						
1	Вказівник на елемент списку	Додавання до списку	Додавання до списку	-	-	-
Функція reconstruct						
1	Вказівник на head та бал нижче середнього	Створення нового списку та виведення	Виведення нового списку	-	-	-
2	Вказівник на head та бал вище середнього	Видалення непотрібних елементів та виведення	Виведення нового списку	-	-	-

Висновок

Я вивчив основні методи та принципи тестування на прикладі створення власної програми