

Звіт по лабораторній роботі № 5

Тема: Розробка програм, що розгалужуються.

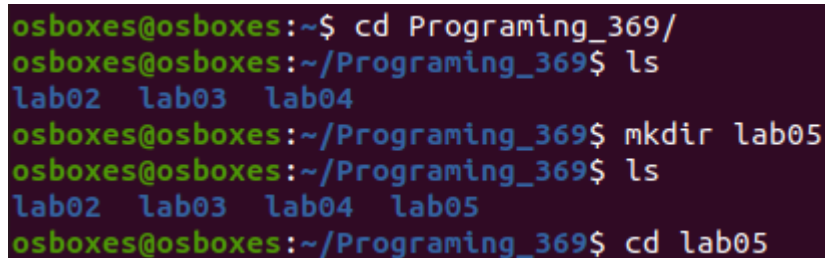
Виконала: студентка 1 курсу групи КІТ-120а Зеленець Олена.

Перевірив: асистент Челак Віктор Володимирович.

Індивідуальне завдання: Визначити, чи є ціле 6-значне число «щасливим» квитком («щасливий квиток» - це квиток, у якому сума першої половини чисел номера дорівнює сумі другої половини).

Опис програми:

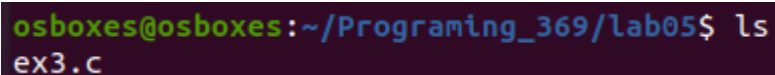
1. Створила у своїй папці «Programing_369» папку «lab05», у якій створюю нові файли.**(Рис.1)**



```
osboxes@osboxes:~$ cd Programing_369/  
osboxes@osboxes:~/Programing_369$ ls  
lab02  lab03  lab04  
osboxes@osboxes:~/Programing_369$ mkdir lab05  
osboxes@osboxes:~/Programing_369$ ls  
lab02  lab03  lab04  lab05  
osboxes@osboxes:~/Programing_369$ cd lab05
```

Рисунок 1 – Створення папки «lab05», у якій будуть усі завдання.

2. Зайшла через команду папо і створила файл «ex3.c», де буде прописаний програмний код згідно мого завдання. **(Рис.2)**



```
osboxes@osboxes:~/Programing_369/lab05$ ls  
ex3.c
```

Рисунок 1- Створений файл із кодом до 3 завдання.

3. Створила “тіло” функції, а саме:

```
«int main(){  
return 0;  
}»
```

5. Тепер оголошуємо наше число через `#define`, яке дорівнює 144306.

6. Далі вказуємо, що наше число `VALUES` є `A` типу `int`.

7. Оголошуємо, що `sum1` і `sum2` (суми перших і останніх трьох цифр) дорівнюють 0.

8. Тепер вводимо перемінну `tic`, яка показуватиме значення кінцевого результату.

9. Далі розпочинаємо перший цикл, який триватиме, поки наше `A=144`:

```
while (A/1000 > 0 ){  
sum1 += A%10; // sum1=6; A = 144306; sum1=6+0; A=14430; sum1 = 6+0+3; A=1443  
A /= 10; // 14430; 1443; 144
```

10. Зараз вводимо нову перемінну `N`, яка дорівнює `A` і розпочинаємо цикл, який виводить останні три цифри щасливого квитка, а саме 306:

```
int N=A;  
while(N > 0 ){  
sum2 += N%10;  
N /= 10;  
}
```

11. І тепер вказуємо значення результату: якщо `sum1==sum2`, то квиток є щасливим, у протилежному випадку не є щасливим квитком

```
if(sum1==sum2) tic = 'Y';  
else tic = 'N';
```

12. Отже, мій програмний код має такий вигляд:

```
int main (){  
    #define VALUES 144306  
    int A = VALUES;  
    int sum1 = 0;  
    int sum2 = 0;  
    char tic;  
    while (A/1000 > 0 ){  
        sum1 += A%10; // sum1=6; A = 144306; sum1=6+0; A=14430; sum1 =  
6+0+3; A=1443  
        A /= 10; // 14430; 1443; 144  
    }  
    int N=A;  
    while(N > 0 ){  
        sum2 += N%10;  
        N /= 10;  
    }  
    if(sum1==sum2) tic = 'Y';  
    else tic = 'N';  
    return 0;  
}
```

13. Даний код на правильність перевіряла в Debug, усе спрацювало і порахувало правильно. Поставила breakpoint після строки з виведенням результату та запустила Debug. Таким чином ми бачимо значення наших змінних після проведення всіх кроків. **(Рис. 3)**

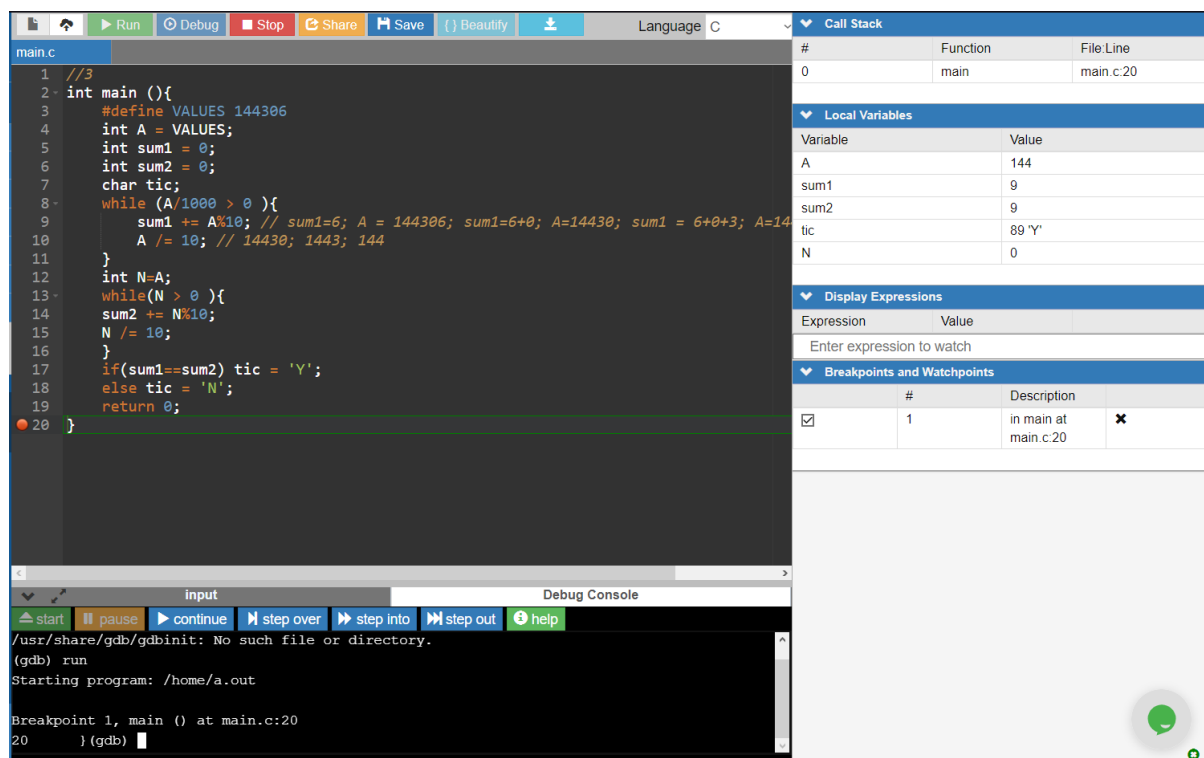


Рисунок 3 – Перевірка правильності коду у Debug, виведення значення результату.

14. Перевіливши, записала цей код у файлі «ex3.c». (Рис. 4)

```
GNU nano 4.8
int main(){
#define VALUES 144306
int A = VALUES;
int sum1 = 0;
int sum2 = 0;
char tic;
while (A/1000 > 0 ){
    sum1 += A%10;
    A /= 10;
}
int N=A;
while(N > 0 ){
    sum2 += N%10;
    N /= 10;
}
if(sum1==sum2) tic = 'Y';
else tic = 'N';
return 0;
}
```

Рисунок 4 – Збережений код через nano.

15. Відкрила командну строку Лінукс, додали зміни до майбутнього коміту, «закомітила» та «запустила» зміни на GitHub у «Programing_369» за допомогою команд: спочатку «git add .», далі «git commit -m "Exercises 1-6"», і врешті «git push».

Висновок:

Отже, ми навчилися розробляти програми за допомогою циклічних конструкцій. Реалізували виконання свого індивідуального завдання, а також інших за допомогою циклів for, while у різних варіаціях.

При виконанні індивідуального завдання навчилися «порівнювати» суми, у нашому випадку, двох частин 6-значного числа.