

Лабораторна робота №7. Функції.

1 Вимоги

1.1. Розробник

1.1.

- Бельчинська Катерина Юріївна
- студентка групи КІТ-320
- 27-nov-2020

1.2. Загальне завдання

- Переробити програми, що були розроблені під час лабораторних робіт з тем “Масиви” та “Цикли” таким чином, щоб для обчислення результату використовувалися функції.
- Реалізувати функцію з варіативною кількістю аргументів.

1.3. Індивідуальне завдання

- Визначити, чи є ціле 6-значне число “щасливим” квитком (сума першої половини чисел номера дорівнює сумі другого).

2 Опис роботи

2.1 Функціональне призначення

- Виконання дій програми за допомогою функцій, для спрощення загального завдання програми. Функція допомагає «розбити» велику задачу на підзадачі, у функції main лише зсилатися на їх опис.

2.2 Варіанти використання

- Для завдання 3 з лабораторної роботи № 5
 - у відладнику netiver викликаємо у функції main три функції, які оприділяли «щасливе» задане число чи ні для циклів for, while do та do while.
 - Викликаємо функцію для числа, обчисленого генератором псевдовипадкових чисел rand():
число «не щасливе»:

ea	Variable	Value	Type
	▼ Local Variables		
	num	845037	int
	sum_half1	17	int
	sum_half2	10	int
	result_for	-1	int
	result_while	-1	int
	result_do	-1	int
	Function Arguments		

Variable	Value	Type
▼ Local Variables		
num	816118	int
sum_half1	15	int
sum_half2	10	int
result_for	-1	int
result_while	-1	int
result_do	-1	int
Function Arguments		

число «ЩАСЛИВЕ»:

Variable	Value	Type
▼ Local Variables		
num	529628	int
sum_half1	16	int
sum_half2	16	int
result_for	1	int
result_while	1	int
result_do	1	int
Function Arguments		

- Для завдання 3 з лабораторної роботи № 6
 - у відладнику `netiver` за допомогою точок зупину заходимо у функцію, яка заповнює результуючий масив заданим числом, перетвореним у строку
 - задане число генерується за допомогою генератора псевдовипадкових чисел `rand()`:

numin	3946
▼ arr	[50]
0	116 't'
1	104 'h'
2	114 'r'
3	101 'e'
4	101 'e'
5	32 ''
6	116 't'
7	104 'h'
8	111 'o'
9	117 'u'
10	115 's'
11	97 'a'
12	110 'n'
13	100 'd'
14	32 ''

numIn	1762
▼ arr	[50]
0	111 'o'
1	110 'n'
2	101 'e'
3	32 ''
4	116 't'
5	104 'h'
6	111 'o'
7	117 'u'
8	115 's'
9	97 'a'
10	110 'n'
11	100 'd'
12	32 ''
13	115 's'
14	101 'e'

numIn	477	int
▼ arr	[50]	char [50]
0	102 'f'	char
1	111 'o'	char
2	117 'u'	char
3	114 'r'	char
4	32 ''	char
5	104 'h'	char
6	117 'u'	char
7	110 'n'	char
8	100 'd'	char
9	114 'r'	char
10	101 'e'	char
11	100 'd'	char
12	32 ''	char
13	115 's'	char
14	101 'e'	char
15	118 'v'	char
16	101 'e'	char

- Структура проекта:

```

├── doc
│   └── lab07.docx
├── Makefile
├── README.md
├── task1_lab05
│   ├── README.md
│   └── src
│       └── main.c
├── task1_lab06
│   ├── README.md
│   └── src
│       └── main.c
├── task2_lab05
│   ├── README.md
│   └── src
│       └── main.c
├── task2_lab06
│   ├── README.md
│   └── src
│       └── main.c
├── task3_lab05
│   └── README.md

```

```
|   └─ src
|       └─ main.c
├─ task3_lab06
|   └─ README.md
|       └─ src
|           └─ main.c
├─ task4_lab05
|   └─ README.md
|       └─ src
|           └─ main.c
├─ task4_lab06
|   └─ README.md
|       └─ src
|           └─ main.c
├─ task5_lab05
|   └─ README.md
|       └─ src
|           └─ main.c
├─ task5_lab06
|   └─ README.md
|       └─ src
|           └─ main.c
└─ task6_lab06
    └─ README.md
        └─ src
            └─ main.c
```

Висновок

В ході даної лабораторної роботи, я навчилася використовувати функції, які повертають і не повертають результат, задля уникання повторів в коді; «спрощувати» загальне завдання, розбиваючи його на менші