

Лабораторна робота №6 Масиви

1 Вимоги

1.1 Розробник

-Носов Микола
-студент групи кіт 1206
-9.11.2020

1.2 Мета:

Навчитися працювати з масивами різних типів даних

1.3 Індивідуальне завдання:

1. Центрувати заданий рядок на площині з із заданим заповнювачем.
Наприклад,
 - заповнювач = “_” ,
 - довжина строки = 15,
 - рядок = "Ivanov \0" (6 символів слово "Ivanov", 8 - пробілів, останній символ = '\0')
 - результат = “____Ivanov____” (4 символи заповнювача, слово "Ivanov", 4 символи заповнювача, останній символ = '\0')
2. Заповнити масив із заданої кількості елементів простими числами, що не повторюються. Розмір вихідного масиву задати наперед відомим значенням, що може будуть більшим ніж результуюча кількість отриманих елементів.
3. Перетворити число (максимальне значення якого - 9999) в рядок.
Наприклад,
 - 123 – “Сто двадцять три”,
 - 4321 – “Чотири тисячі триста двадцять один”.
4. У заданому тексті знайти кількість слів за умови, що між словами може бути будь-яка кількість пропусків.
5. Дано двовимірний масив з $N \times N$ цілих чисел. Виконати циклічне зрушення елементів рядків масиву в напрямку справа наліво (перший елемент рядка повинен переміститися в її кінець).
6. Дано двовимірний масив з $N \times N$ цілих чисел. Помножити матрицю саму на себе (відповідно до правил множення матриць).

2 Опис програми

2.1 Функціональне призначення

Програма призначена для;

- центрування слів у строчці
- знаходження простих, неповторюваних чисел
- перетворення числа в рядок
- знаходження кількості слів у тексті
- виконання циклічного зрушення наліво матриці
- знаходження квадрата матриці

2.2 Логічна структура

- макроси (див Рис 2.1)
- головна функція main
- структура проєкту (див. Рис.2.2)

```
#define LEN 15 /*Макрос для 1 задачі */  
#define VALUE 10 /*Макрос для 3 задачі */
```

Рисунок 2.1 - початкові дані

```
├── dist  
│   └── lab06.bin  
├── doc  
├── Makefile  
├── README.md  
├── src  
│   └── lab06.c  
3 directories, 4 files
```

Рисунок 2.2 - структура

2.3 Важливі фрагменти коду

- прив'язка остачі від ділення числа до номеру потрібної рядка у масиві (див рис 2.4)
- обробка пропусків між словами (див рис 2.5)
- обробка виключень (11, 12,13....., 19) (див ри 2.6)
- обробка пропусків перед словом та після слова (див рис 2.7)
- множення матриць (див рис 2.8)

```
if (VALUE >= 1 && VALUE % 10 != 0)
{
    result[3] = DIGIT[(VALUE % 10)];
}
```

Рисунок 2.4 - ділення

```
for (int i = 0; i < (LEN - 1); i++)
{
    if (mass[i] != '\0' && mass[i] != ' ')
    {
        shetBukv++;
    }
    if (mass[i] == ' ' && mass[i+1] != ' ' && mass[i-1] != ' ') /* обработка пробела между словами */
    {
        shetBukv++;
    }
}
```

Рисунок 2.5 - обробко пропусків

```
if (VALUE % 10 + (VALUE / 10 % 10) * 10 == i)
{
    result[3] = '\0';
    result[2] = EXCEPTIONS[i%10 - 1];
}
}
```

Рисунок 2.6 - обробка виключення

```
}
if (text1[size - 2] != ' ') /* если строка заканчивается пробелом. -2 так как sizeof считает " за элемент */
{
    chet1++;
}
if (text1[0] == ' ') /* если строка начинается с пробела */
{
    chet1--;
}
```

Рисунок 2.7 - обробка пропусків

```

for (int i1 = 0; i1 < N1; i1++)/* Умножение по правилам математики */
{
    for (int j1 = 0; j1 < N1; j1++)
    {
        for (int q1 = 0; q1 < N1; q1++)
        {
            sum = sum + mas1[q1][i1] * mas1[j1][q1];

        }
        res[j1][i1] = sum;
        sum = 0;
    }
}

```

Рисунок 2.8 - множення матриць

3 Варіанти використання

Для обчислення результатів ми використовуємо gdb - дебагер, влаштований в утиліті gcc.

Щоб подивитися на результат, ми повинні при компіляції вказати рівень інформації для відладки за допомогою -o, відкомпілювати код, запустити його бінарний файл у відлагоднику та вказати головну функцію, як місце, з якого починати відладку за допомогою команди “b main”, запустити програму та дивитись за результатом її роботи.

Початкові данні :

LEN = 15;

VALUE = 1;

-Відценроване слово (див рис 3.1)

-Массив простиx чисел (див рис 3.1)

-Перетворення числа на рядок (див рис 3.2)

-Кількість слів у рядку (див рис 3.3)

-Сдвиг матриці (див рис 3.4)

-Множення матриці (див рис 3.5)

```

centr = 2
mass = "    Ivan Borov  "
shetSlov = 10
firstChar = 3
newmass = "__Ivan Borov__"
mass1 = {3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53}

```

Рисунок 3.1 - значення мінних у gdb, центрування та массив

```

result = {0x0, 0x0, 0x0, 0x555555557061 "один"}

```

Рисунок 3.2 - значення у gdb, переведення

```
text1 = "Спасибо    за      отличные    выходные"  
chet1 = 4
```

Рисунок 3.3 - значення у gdb, обчислення кількості слів

```
mas = {{6, 7, 3}, {3, 5, 5}, {2, 9, 6}}  
resmas = {{3}, {5}, {6}}
```

Рисунок 3.4 - значення у gdb, результат сдвигу

```
mas1 = {{1, 2, 7}, {0, 9, 3}, {6, 0, 6}}  
res = {{43, 20, 55}, {18, 81, 45}, {42, 12, 78}}
```

Рисунок 3.5 - значення у gdb, результат множення

Висновок :

В умовах даної лабораторної роботи ми набули практичного досвіду з роботою з масивами