

Лабораторна робота № 6. Масиви

1 ВИМОГИ

1.1 Розробник

- Фоменко Гліб Володимирович;
- студент групи КІТ-121Г;
- 08-гру-2021.

1.2 Загальне завдання

Розробити програму використовуючи функції

1.3 Індивідуальне завдання

Сформувати функцію, що переводить число в речення з слів. Ліміт числа: від -9999 до 9999

2 ОПИС ПРОГРАМИ

2.1 Функціональне призначення

Програма призначена для переведення цілого числа в речення (рядок)

При знаходінні числа діють наступні обмеження:

- число яке передається в функцію повинно бути від -9999 до 9999
- число повинно бути цілим

Результат зберігається у змінній result.

Подивитися результат виконання можна тільки в режимі дебагінгу

2.2 Опис логічної структури

Спочатку програма бере число (випадкове або взятє з аргументу командної строки), кожен цифру цього числа записує в змінні, проходить по розміру масиву (кількість можливих слів) та в залежності від того яке зараз за рахунком слово і яка цифра в поточній змінній, записує в масив слів, слово що відповідає поточній змінній. Далі програма переписує масив слів в масив char (символів, результуючий масив). Видаляє зайві пробіли в результуючому масиві.

2.2.1 Функція переведення числа в речення

```
void make_string_from_number(int number, char printableResult[])
```

Призначення: переведення числа в речення.

Опис роботи: бере число що було передано як аргумент в функцію, створює речення на основі цього числа та передає його в printableResult[].

2.2.2 Основна функція

```
int main(int argc, char **argv)
```

Призначення: головна функція.

Опис роботи:

- якщо через аргументи командної строки нічого не передається, то для аргументу number створюються випадкове число.
- число, яке було передано через аргумент строки або випадкове, передається в функцію make_string_from_number.
- результат виконання записується в змінну result.

2.2.3 Структура проекту

lab06

```
├── Doxyfile
├── Makefile
├── doc
│   ├── lab06.docx
│   ├── lab06.md
│   └── lab06.pdf
└── src
    └── main.c
```

2.3 Важливі фрагменти програми

2.3.1 Запис в змінну кожної цифри числа

```
int firstFigure = number / 1000;
```

```
int secondFigure = (number / 100) % 10;
int thirdFigure = (number / 10) % 10;
int fourthFigure = number % 10;
```

2.3.2 Приклад заповнення масиву словами

case 1:

```
if (firstFigure > 0) {
    for (int j = 0; j < STRING_SIZE; j++) {
        result[i][j] = figures[firstFigure][j];
    }
    i++;
    for (int j = 0; j < (int)sizeof(thousand); j++) {
        result[i][j] = thousand[j];
    }
}
break;
```

3 ВАРІАНТ ВИКОРИСТАННЯ

Для демонстрації результатів задачі використовується:

- покрокове виконання програми в утиліті lldb;

Варіант використання: послідовність дій для запуску програми у режимі відлагодження:

- запустити програму у відлагоднику lldb;
- поставити точку зупинки на функції main (строка з `return 0;`);
- запустити програму;
- подивитись результати виконання програми, оглядаючи значення змінної animals

Process 25088 stopped

* thread #1, name = 'main.bin', stop reason = breakpoint 1.1

```

frame #0: 0x00000000004011fb main.bin`main(argc=2, argv=0x00007ffffffe2f8)
at main.c:82:2
    79      char result[RESULT_COUNT];
    80      make_string_from_number(number, result);
    81
-> 82      return 0;
    83  }
    84
    85  void make_string_from_number(int number, char printableResult[])
(lldb) p result
(char [100]) $0 = "four hundred and thirty nine"

```

ВИСНОВКИ

При виконанні даної лабораторної роботи було набуто практичного досвіду роботи із рядками в си (char масивами). Наприклад те що, щоб в рядок не "затікало" сміття, треба його обмежити символом - '\0'