

Кожевніков Ілля КН-923в
Лабораторна робота №6
Масиви

Обрав 23. (**) Перетворити число (максимальне значення якого - 9999) в рядок. (усі символи нижнього регістру на виході).

Наприклад,

- 123 – “one hundred twenty three”,
- 4311 – “four thousands three hundreds eleven”

```
kozhevka@AbobaLaptop:~/programming-kozhevnikov/lab06/build$ ./main.bin
Введіть число в межах від 0 до 9999: 3219
three thousand two hundred nineteen
kozhevka@AbobaLaptop:~/programming-kozhevnikov/lab06/build$ ./main.bin
Введіть число в межах від 0 до 9999: 6479
six thousand four hundred seventy nine
kozhevka@AbobaLaptop:~/programming-kozhevnikov/lab06/build$ ./main.bin
Введіть число в межах від 0 до 9999: 9999
nine thousand nine hundred ninety nine
kozhevka@AbobaLaptop:~/programming-kozhevnikov/lab06/build$ ./main.bin
Введіть число в межах від 0 до 9999: 99999
Помилка умов! Введіть число в межах від 0 до 9999:
Введіть число в межах від 0 до 9999: -99999
Помилка умов! Введіть число в межах від 0 до 9999:
Введіть число в межах від 0 до 9999: 1220
one thousand two hundred twenty
kozhevka@AbobaLaptop:~/programming-kozhevnikov/lab06/build$ ./main.bin
Введіть число в межах від 0 до 9999: 2008
two thousand eight
kozhevka@AbobaLaptop:~/programming-kozhevnikov/lab06/build$ |
```

1. Ініціював масиви з назвами чисел,
Для простих чисел
Для 11-19
Для десятків
Та Сотень і Тисяч

```
// Масиви рядків для словесного представлення чисел
char *units[] = { [0]: "", [1]: "one", [2]: "two", [3]: "three", [4]: "four", [5]: "five", [6]: "six", [7]: "seven", [8]: "eight", [9]: "nine" };
char *teens[] = { [0]: "", [1]: "eleven", [2]: "twelve", [3]: "thirteen", [4]: "fourteen", [5]: "fifteen", [6]: "sixteen", [7]: "seventeen", [8]: "eighteen", [9]: "nineteen" };
char *tens[] = { [0]: "", [1]: "ten", [2]: "twenty", [3]: "thirty", [4]: "forty", [5]: "fifty", [6]: "sixty", [7]: "seventy", [8]: "eighty", [9]: "ninety" };
char *hundreds[] = { [0]: "hundred", [1]: "thousand" };
```

2. Повторив метод для вводу цілого числа(з минулих лаб) та змінив повідомлення помилки.

```
int numberInput(const char* message)
{
    int result;

    int success = 0;

    while (!success) {
        printf("%s", message);
        if (scanf("%d", &result) == 1)
        {
            if (result > 9999 || result < 0)
            {
                printf("Помилка умов! Введіть число в межах від 0 до 9999:\n");
                continue;
            }
            success = 1;
        }
        else
        {
            printf("Помилка формату! Введіть коректне число.\n");
            while (getchar() != '\n');
        }
    }

    return result;
}
```

- Після вводу числа користувачем,
Ініціалізується масив розміром з кількість чисел у максимально допустимому числі,
Та до кожного індексу в масиві присвоюється його число

```
int main() {  
    int number;  
  
    // Введення числа  
    number = numberInput( message: "Введіть число в межах від 0 до 9999: ");  
  
    int n = number; //копіюємо число для математичних операцій  
  
    int numberArray[4];  
  
    n = number;  
  
    //Присвоїти число кожному індексу  
  
    for (int i = 3; i >= 0; i--)  
    {  
        numberArray[i] = n % 10; //Присвоюємо цифру індексу в масиві  
        n /= 10;  
    }  
}
```

4. Далі перевіряємо кожен індекс масиву на наявність числа у ньому і даємо назву. (Можна було б загорнути у один цикл та обробляти більш ніж 4х значні числа, додавши до масиву сотень-тисяч ще мільйон та далі, але умовою були тільки 4х значні числа тож вирішив гаяти час))
І видаємо результат у консоль.

```
char result[50] = ""; // Рядок для зберігання словесного представлення числа

if (numberArray[0] != 0) // Якщо у числі є тисячі
{
    sprintf(result, "%s %s", units[numberArray[0]], hundreds[1]);
}

if (numberArray[1] != 0) //Якщо у числі є сотні
{
    sprintf(result, "%s %s %s", result, units[numberArray[1]], hundreds[0]);
}

if (numberArray[2] >= 2) //Якщо десятки більше 20ти включно
{
    sprintf(result, "%s %s %s", result, tens[numberArray[2]], units[numberArray[3]]);
}
else
{
    if (numberArray[2] * 10 + numberArray[3] > 10) // Якщо десятки в межах 11-19
    {
        sprintf(result, "%s %s", result, teens[numberArray[3]]);
    } else // немає десятків
    {
        sprintf(result, "%s %s", result, units[numberArray[3]]);
    }
}

printf("%s\n", result);
return 0;
```

1. Як рядки представляються в мові програмування C?

Рядки в C представлені масивами символів, де кожен символ є елементом масиву. Рядок завершується нульовим символом '\0'.

2. Що таке “мірність масиву”? Скільки вимірів може мати масив?

"Мірність масиву" вказує на кількість індексів, які використовуються для доступу до елементів. Масив може бути одномірним, двовимірним, тривимірним, і так далі. Кількість вимірів визначається кількістю індексів, використовуваних для доступу.

3. Від чого залежить об'єм пам'яті, що необхідний для зберігання елементів масиву?

Об'єм пам'яті, необхідний для зберігання елементів масиву, залежить від розміру типу даних, які складають масив, та від кількості елементів у масиві.

4. Як можна звернутися до елемента масиву?

Елемент масиву звертається за допомогою індексу, наприклад, `array[index]`, де `array` - назва масиву, а `index` - індекс елемента.

5. Як оголошуються масиви?

Масив оголошується за допомогою вказання типу даних, ім'я масиву та розміру (якщо це оголошення масиву з визначеним розміром) або за допомогою ініціалізації (без вказання розміру).

6. Дайте оцінку наступному оголошенню масиву:

```
int mas[] = { 1, 2, 3, 4, 5, 6, 7, 8 };
```

`int mas[] = { 1, 2, 3, 4, 5, 6, 7, 8 };` - це оголошення одномірного масиву `mas` з 8 елементами, і йому призначається розмір 8, оскільки розмір визначається автоматично.

7. При доступі до елементів двовимірного масиву укажіть, яке призначення першого та другого індексів?

Перший індекс вказує на рядок (підмасив), а другий - на стовпець (елемент у масиві рядка).

8. Дайте оцінку наступному оголошенню масиву:

```
int mas[][2] = {  
    { 1, 2 },  
    { 3, 4 },  
    { 5, 6 },  
    { 7, 8 }  
};
```

це оголошення двовимірного масиву `mas` з 4 рядками та 2 стовпцями.

9. В якому порядку зберігаються у пам'яті комп'ютера елементи двовимірного масиву?

Елементи зберігаються по рядках. Спочатку елементи одного рядка, потім іншого рядка і так далі.

10. Чим характеризуються діагональні елементи багатовимірних масивів? Скільки операторів циклу необхідно для їх обробки?

Діагональні елементи - це ті, які розташовані вздовж діагоналі масиву. Кількість операторів циклу для їх обробки залежить від кількості вимірів у масиві. Для двовимірного масиву, наприклад, зазвичай потрібно два оператори циклу.