

ДОСЛІДЖЕННЯ ОСОБЛИВОСТЕЙ ФОРМАТІВ ТА ТИПІВ ДАНИХ СУЧАСНИХ КОМПІЛЯТОРІВ

У мові Сі немає спеціального типу даних який би визначав символічний тип. Тип даних char є цілочисельним, таким як int, short тощо. Char є найменшим цілим типом (займає 1 байт) і здебільшого призначений для представлення символів. Наприклад, кодування символу «а», коли «а» вважається частиною набору символів ASCII, є 97 у десятковій системі числення, або 01100001 у двійковій системі числення. Якщо символ «а» належить до іншого набору символів, він може мати інше кодування [1]. Таким чином, як і будь-який інший цілий тип, він може бути зі знаком або без знака. Тобто окрім char може бути signed char і unsigned char.

Якщо тип char є беззнаковий, то він може містити тільки додатні значення, а його діапазон, визначений стандартом С, знаходиться в межах від 0 до 255. Якщо тип char є знаковим, то він може містити 0, від'ємні та додатні значення, а його діапазон, визначений стандартом С, знаходиться в межах від -127 до 127. Усі три типи різні, але вони мають однаковий розмір.

Для прикладу вибираємо тип даних signed char. При введенні числа яке не попадає в діапазон типу даних signed char ([-128;127]), але попадає у восьмібітний знаковий формат, введене число буде опрацьовуватися компілятором Visual Studio 2022 за наступним алгоритмом:

1. Зберігаємо знаковий біт. Від двійкового коду записаного в решті бітів віднімаємо одиницю та інвертуємо. Отримуємо результуюче знакове значення.

Наприклад: вводимо значення 150. Використовуємо даний алгоритм.

$$\begin{array}{r} 1|0010110 \\ | \quad -1 \\ \hline 1|1101010 \end{array}$$

Отримуємо значення – 106. Результат виконання програми зображено на рисунку 1.

```
Введіть значення з діапазону [-128..127]  
a = 150  
a = -106
```

Рис 1. Результат виконання програми, введено значення 150

При введенні числа яке не попадає в діапазон типу даних signed char([-128;127]), та не попадає в восьмібітний формат, введене число буде опрацьовуватися компілятором Visual Studio 2022 за таким алгоритмом:

2. Відсікаємо молодші біти до потрібного нам розміру та використовуємо попередній алгоритм, але з однією умовою:

- a. якщо біт який припадає на знаковий біт дорівнює одиниці то використовуємо попередній алгоритм;
- b. якщо біт який припадає на знаковий біт дорівнює нулю, то наше число буде дорівнювати числу записаному в 7 бітах та це значення буде беззнакове.

Наприклад: вводимо значення 400. Відсікаємо молодший біт **110010000**. Біт який припадає на знаковий біт дорівнює одиниці тому використовуємо алгоритм (1).

$$\begin{array}{r} 1|0010000 \\ | \quad -1 \\ \hline 1|1110000 \end{array}$$

Отримуємо значення -112. Результат виконання програми зображено на рисунку 2.

```
Введіть значення з діапазону [-128..127]  
a = 400  
a = -112
```

Рис 2.Результат виконання програми, введено значення 400

Вводимо значення 300. Відсікаємо молодший біт. 100101100. Біт який припадає на знаковий біт дорівнює нулю тому наше число 0101100=44.

```
Введіть значення з діапазону [-128..127]  
a = 300  
a =44
```

Рис 3.Результат виконання програми, введено значення 300

В дослідженні розглянуто тип даних signed char і що відбувається, якщо введено число не потрапляє в діапазон даних зазначеного типу.

Список використаних джерел

1. Wael M. What is char , signed char , unsigned char , and character literals in C? – 2020. – URL: <https://medium.com/analytics-vidhya/what-is-char-signed-char-unsigned-char-and-character-literals-in-c-796034139b98>.