

Задание ВІТ

Манипуляции с битами

В следующих задачах предлагается написать ту или иную функцию для преобразования целых чисел, оптимально используя битовые операции языка Си. При разработке алгоритма и написании функции не следует заранее делать никаких предположений о размере целых чисел в битах, пользуясь при необходимости операцией `sizeof()`.

Для того чтобы обобщить свой код, абстрагироваться от конкретного целого типа (который в зависимости от вашего выбора, архитектуры системы, компилятора может иметь разный размер в битах), **введём свой собственный тип** с именем `integer_t`. Определим его как синоним одного из имеющихся беззнаковых типов языка Си при помощи ключевого слова `typedef`:

```
// typedef unsigned char  integer_t;
// typedef unsigned short integer_t;
typedef unsigned int      integer_t;
// typedef unsigned long  integer_t;
```

Не только написанная функция, но и вся программа должна быть целиком универсальной относительно конкретного определения типа `integer_t`. Сразу следует предостеречь от стандартной ошибки: проверок вида `if (sizeof(integer_t) == ...)` в коде быть не должно.

Требуется использовать эту функцию в тестовой программе, которая должна читать с клавиатуры числа и давать ответ до тех пор, пока пользователь не введёт ноль (перед выходом из программы **для нуля ответ тоже должен выдаться**). Введённое значение и полученный результат необходимо выводить на экран **в трёх системах счисления**: десятичной, двоичной и шестнадцатеричной, полностью указывая **даже ведущие нулевые цифры** в последних двух системах, дополняя число до максимально возможного количества цифр. Прототип функции: `integer_t Process(integer_t n);`

Комментарии к задаче

Форматирование вывода Данные входа и выхода должны располагаться один на другом, как показано в примерах. Кроме того, должно иметь место выравнивание данных, то есть каждое поле находится над (под) соответствующим ему.

В случае вывода в шестнадцатеричной и двоичной системах этого эффекта достигнуть легко - нужно выводить числа с максимально доступной для данного типа шириной. Для шестнадцатеричной системы этого эффекта можно добиться с помощью модификатора ширины (см. задачу SER). Для того, чтобы ведущие символы заполнялись не пробелами, а нулями, необходимо указать символ "0" (например, `%03i`).

В случае с двоичной системы, функция вывода реализуется самим пользователем, в ней он выводит значения всех бит всех байт ячейки типа `integer_t`.

Немного более сложная задача стоит для вывода чисел в десятичной системе счисления. Они должны выводиться по максимальной длине одного из чисел (как показано в варианте BIT-3). Для достижения данного эффекта можно использовать флаг ширины функции `printf()`, но указывать не число, а символ "*", в этом случае ширина будет считываться из параметров, например код `printf("%*i", 8, var);` напечатает значение переменной `var` с шириной 8.

Единообразное считывание чисел разного типа Различные варианты типа `integer_t` имеют разный размер. При этом, выводиться считываться они должны с помощью единого фрагмента кода. В случае использования одного и того же флага, будут возникать переполнения, если тип имеет слишком маленький размер или недобор данных в обратном случае. Для решения этой проблемы рекомендуется ввести еще один тип `MaxInt`. Размер этого типа равен размеру максимального из используемых. Далее, в коде программы все считывание и вывод производится **только** в переменные этого типа. А для работы с переменной типа `integer_t` производятся соответствующие преобразования типов (туда и обратно).

План решения

1. Для начала необходимо разобраться, какие инструменты потребуются для решения задачи. Необходимо понять, как выставлять в нужное значение некоторый бит или байт числа.
2. Далее, требуется разобраться, как отличаются друг от друга разные варианты типов `integer_t` и как, используя один и тот же код, работать со всеми типами одинаково. После этого можно приступить к реализации основной функции. Тестирование, в целях ускорения работы, рекомендуется проводить на статически заданных в коде программы числах (передавая их в свои функции).

3. После того, как основная часть реализована и протестирована, нужно написать считывание чисел с клавиатуры, а так же вывод с необходимым форматированием.

Варианты

Вариант ВІТ-1 (Реверс битов). В рамках общего условия задачи написать функцию, которая по заданному числу типа `integer_t` возвращает другое число, в котором переставлены местами самый младший бит с самым старшим, второй сверху со вторым снизу и т.д. Пример работы для 8-битного типа `char`:

Задание ВІТ-1: Реверс битов (8 бит)

Введите число: 163

Вы ввели : 163 = 0xA3 = 10100011

Результат : 197 = 0xC5 = 11000101

Введите число: _

Вариант ВІТ-2 (Реверс пар). В рамках общего условия задачи написать функцию, которая по заданному числу типа `integer_t` возвращает другое число, в котором переставлены местами пары битов: самая младшая пара с самой старшей, вторая сверху со второй снизу и т.д. Пример работы для 8-битного типа `char`:

Задание ВІТ-2: Реверс пар (8 бит)

Введите число: 163

Вы ввели : 163 = 0xA3 = 10100011

Результат : 202 = 0xCA = 11001010

Введите число: _

Вариант ВІТ-3 (Реверс четверок). В рамках общего условия задачи написать функцию, которая по заданному числу типа `integer_t` возвращает другое число, в котором переставлены местами четверки битов: самая младшая четверка с самой старшей, вторая сверху со второй снизу и т.д. Пример работы для 8-битного типа `char`:

Задание ВІТ-3: Реверс четверок (8 бит)

Введите число: 163

Вы ввели : 163 = 0xA3 = 10100011

Результат : 58 = 0x3A = 00111010

Введите число: _

Вариант ВІТ-4 (Реверс байтов). В рамках общего условия задачи написать функцию, которая по заданному числу типа `integer_t` возвращает другое число, в котором переставлены местами байты: самый младший байт с самым старшим, второй сверху со вторым снизу и т.д. Пример работы для 16-битного типа `short`:

Задание ВІТ-4: Реверс байтов (16 бит)

Введите число: **41906**

Вы ввели : 41906 = 0xA3B2 = 10100011 10110010

Результат : 45731 = 0xB2A3 = 10110010 10100011

Введите число: _

Вариант ВІТ-5 (Обмен битов). В рамках общего условия задачи написать функцию, которая по заданному числу типа `integer_t` возвращает другое число, в котором переставлены местами соседние биты: первый со вторым, третий с четвертым и т.д. Пример работы для 8-битного типа `char`:

Задание ВІТ-5: Обмен битов (8 бит)

Введите число: **163**

Вы ввели : 163 = 0xA3 = 10100011

Результат : 83 = 0x53 = 01010011

Введите число: _

Вариант ВІТ-6 (Реверс битов в четверках). В рамках общего условия задачи написать функцию, которая по заданному числу типа `integer_t` возвращает другое число, в котором переставлены местами биты в четверках: самый младший бит четверки с самым старшим битом той же четверки, второй сверху со вторым снизу и т.д. Пример работы для 8-битного типа `char`:

Задание ВІТ-6: Реверс битов в четверках (8 бит)

Введите число: **163**

Вы ввели : 163 = 0xA3 = 10100011

Результат : 92 = 0x5C = 01011100

Введите число: _

Вариант ВІТ-7 (Реверс битов в байтах). В рамках общего условия задачи написать функцию, которая по заданному числу типа `integer_t` возвращает другое число, в котором переставлены местами биты в байтах: самый младший бит байта с самым старшим битом того же байта, второй сверху со вторым снизу и т.д. Пример работы для 16-битного типа `short`:

Задание ВІТ-7: Реверс битов в байтах (16 бит)

Введите число: **23456**

Вы ввели : 23456 = 0x5BA0 = 01011011 10100000

Результат : 55813 = 0xDA05 = 11011010 00000101

Введите число: _

Вариант ВІТ-8 (Циклический сдвиг влево). В рамках общего условия задачи написать функцию, которая по заданному числу типа `integer_t` и целому числу возвращает результат циклического сдвига первого числа влево на количество бит, определяемое вторым числом. Пример работы для 16-битного типа `short`:

Задание ВІТ-8: Циклический сдвиг влево (16 бит)

Введите числа: **23456 3**

Вы ввели : 23456 = 0x5BA0 = 01011011 10100000

Результат : 56578 = 0xDD02 = 11011101 00000010

Введите числа: _

Вариант ВІТ-9 (Циклический сдвиг вправо). В рамках общего условия задачи написать функцию, которая по заданному числу типа `integer_t` и целому числу возвращает результат циклического сдвига первого числа вправо на количество бит, определяемое вторым числом. Пример работы для 16-битного типа `short`:

Задание ВІТ-9: Циклический сдвиг вправо (16 бит)

Введите числа: **23456 3**

Вы ввели : 23456 = 0x5BA0 = 01011011 10100000

Результат : 2932 = 0x0B74 = 00001011 01110100

Введите числа: _

Вариант BIT-10 (Циклический сдвиг влево внутри байтов). В рамках общего условия задачи написать функцию, которая по заданному числу типа `integer_t` и целому числу возвращает результат, в котором каждый байт циклически сдвинут влево на количество бит, определяемое вторым числом. Пример работы для 16-битного типа `short`:

Задание BIT-10: Циклический сдвиг влево внутри байтов (16 бит)

Введите числа: **23456 3**

Вы ввели : 23456 = 0x5BA0 = 01011011 10100000

Результат : 55813 = 0xDA05 = 11011010 00000101

Введите числа: __

Вариант BIT-11 (Циклический сдвиг вправо внутри байтов). В рамках общего условия задачи написать функцию, которая по заданному числу типа `integer_t` и целому числу возвращает результат, в котором каждый байт циклически сдвинут вправо на количество бит, определяемое вторым числом. Пример работы для 16-битного типа `short`:

Задание BIT-11: Циклический сдвиг вправо внутри байтов (16 бит)

Введите числа: **23456 3**

Вы ввели : 23456 = 0x5BA0 = 01011011 10100000

Результат : 27412 = 0x6B14 = 01101011 00010100

Введите числа: __