

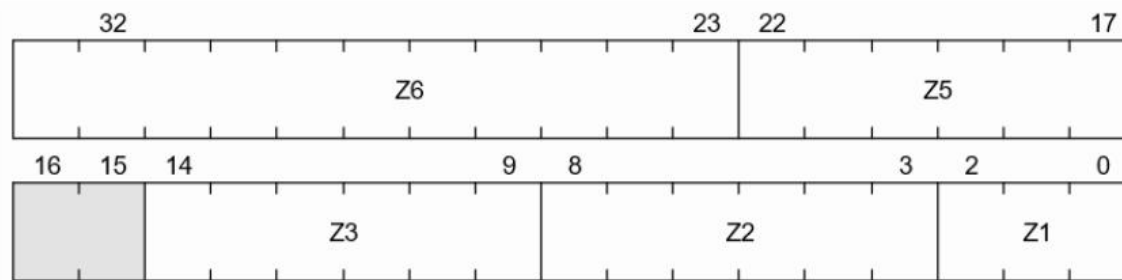
1. Реализовать функцию для кодирования данных, содержащих битовые поля. В решении необходимо использовать побитовые операции. Неиспользуемые поля результата должны содержать нулевые биты.

**Входные данные:**

Словарь из битовых полей. Значения битовых полей имеют тип: десятичная строка.

**Выходные данные:**

Шестнадцатиричная строка.



**Тесты:**

```
>>> main({'z1': '3', 'z2': '2', 'z3': '11', 'z5': '32', 'z6': '525'})
'0x106c01613'

>>> main({'z1': '5', 'z2': '55', 'z3': '2', 'z5': '53', 'z6': '1016'})
'0x1fc6a05bd'

>>> main({'z1': '3', 'z2': '46', 'z3': '43', 'z5': '29', 'z6': '387'})
'0xc1ba5773'

>>> main({'z1': '3', 'z2': '47', 'z3': '51', 'z5': '40', 'z6': '550'})
'0x11350677b'
```

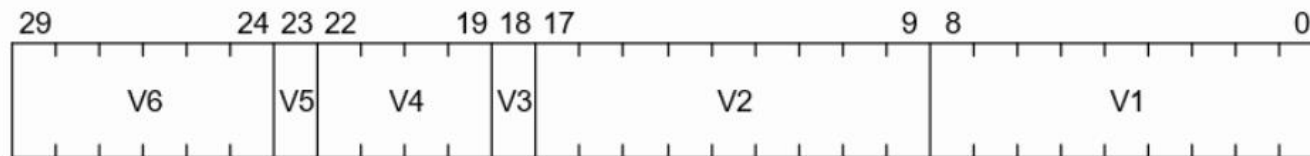
2. Реализовать функцию для кодирования данных, содержащих битовые поля. В решении необходимо использовать побитовые операции.

**Входные данные:**

Кортеж из битовых полей в порядке от младших бит к старшим. Значения битовых полей имеют тип: целое.

**Выходные данные:**

Целое.



*Тесты:*

```
>>> main((376, 159, 1, 0, 1, 21))
361054072

>>> main((130, 60, 1, 11, 1, 52))
886864002

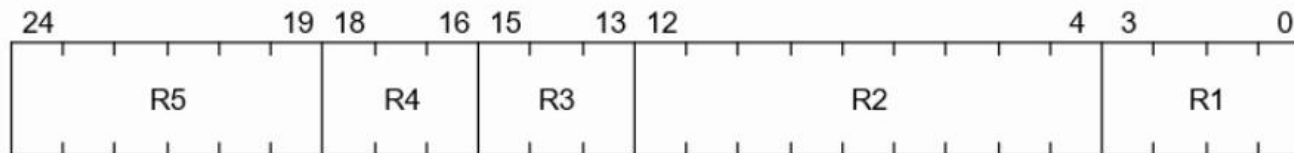
>>> main((396, 444, 0, 14, 0, 33))
561215884

>>> main((110, 199, 1, 12, 0, 27))
459640430
```

3. Реализовать функцию для декодирования данных, содержащих битовые поля. В решении необходимо использовать побитовые операции.

**Входные данные:**

Шестнадцатиричная строка.



**Выходные данные:**

Список из битовых полей в виде пар имя-значение. Значения битовых полей имеют тип: десятичная строка.

*Тесты:*

```
>>> main('0x6a9f6')
[('R1', '6'), ('R2', '159'), ('R3', '5'), ('R4', '6'), ('R5', '0')]

>>> main('0x2464d1')
[('R1', '1'), ('R2', '77'), ('R3', '3'), ('R4', '4'), ('R5', '4')]

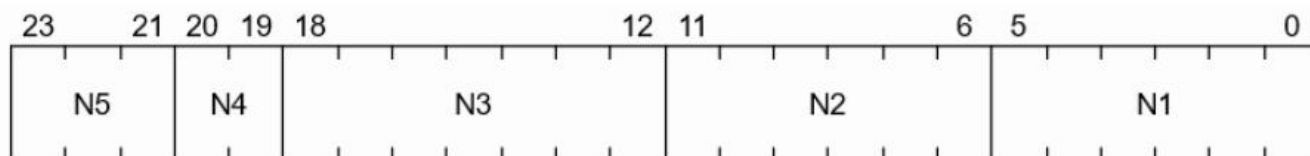
>>> main('0x18ba48b')
[('R1', '11'), ('R2', '72'), ('R3', '5'), ('R4', '3'), ('R5', '49')]

>>> main('0xb077e5')
[('R1', '5'), ('R2', '382'), ('R3', '3'), ('R4', '0'), ('R5', '22')]
```

4. Реализовать функцию для декодирования данных, содержащих битовые поля. В решении необходимо использовать побитовые операции.

**Входные данные:**

Шестнадцатиричная строка.



**Выходные данные:**

Словарь из битовых полей. Значения битовых полей имеют тип: целое.

*Тесты:*

```
>>> main('0xb9a562')
{'N1': 34, 'N2': 21, 'N3': 26, 'N4': 3, 'N5': 5}
```

```
>>> main('0x85c1a5')
{'N1': 37, 'N2': 6, 'N3': 92, 'N4': 0, 'N5': 4}
```

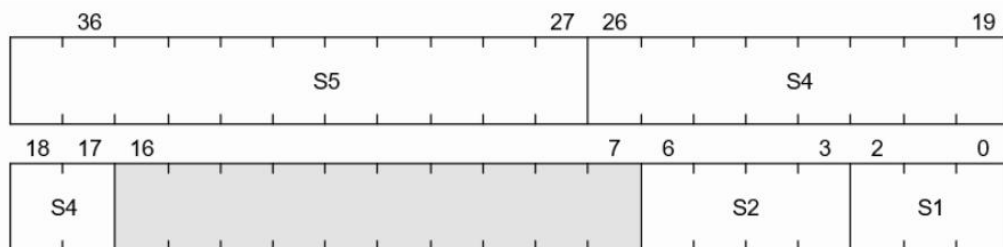
```
>>> main('0xe3fb96')
{'N1': 22, 'N2': 46, 'N3': 63, 'N4': 0, 'N5': 7}
```

```
>>> main('0x2909e9')
{'N1': 41, 'N2': 39, 'N3': 16, 'N4': 1, 'N5': 1}
```

5. Реализовать функцию для транскодирования данных, содержащих битовые поля. В решении необходимо использовать побитовые операции. Неиспользуемые поля результата должны содержать нулевые биты.

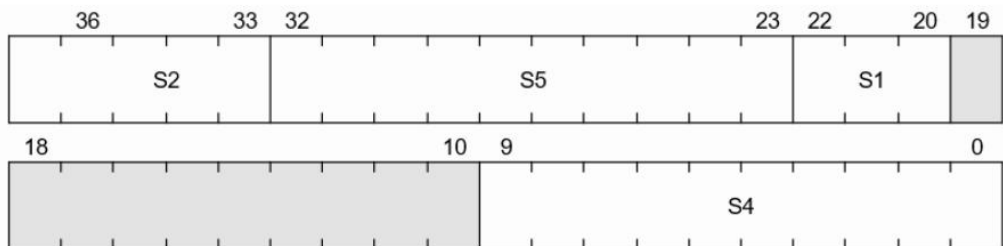
**Входные данные:**

Десятичная строка.



**Выходные данные:**

Шестнадцатиричная строка.



**Тесты:**

```
>>> main('239948252')
'0x1600c00326'
```

```
>>> main('128390864606')
'0x17de600258'
```

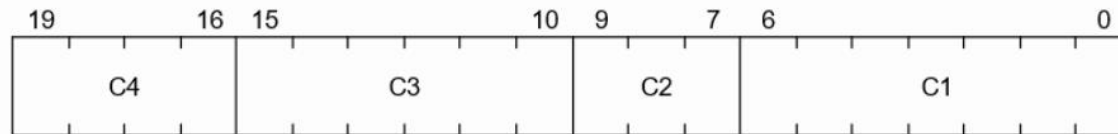
```
>>> main('51700518902')
'0x1cc0e000cb'
```

```
>>> main('94668019137')
'0x1160900153'
```

6. Реализовать функцию для транскодирования данных, содержащих битовые поля. В решении необходимо использовать побитовые операции.

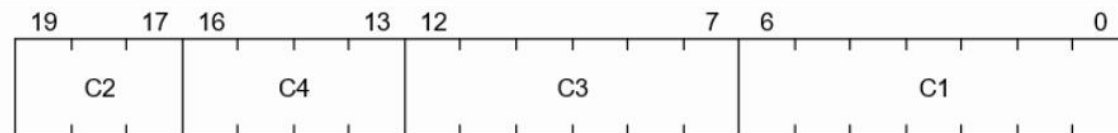
**Входные данные:**

Десятичная строка.



**Выходные данные:**

Десятичная строка.



**Тесты:**

```
>>> main('743096')  
'748216'
```

```
>>> main('42860')  
'791788'
```

```
>>> main('742832')  
'486064'
```

```
>>> main('992703')  
'517311'
```