

```

build = '1.00'
print('\n', 'build:', build, '\n')

def main_menu():
    print('Выбери нужный тебе калькулятор (впиши цифру) \n'
          '1) Обычный калькулятор \n'
          '2) Калькулятор масс \n'
          '3) Калькулятор валют \n'
          '4) Калькулятор ипотеки \n'
          '5) Калькулятор температуры \n'
          '6) Калькулятор логических выражений')
    number_of_calc = input('Номер:')
    return number_of_calc

def check(text_menu, value):
    print(text_menu)
    user_n = input()
    if not (user_n.isdigit() and int(value) >= int(user_n) >= 1):
        return check(text_menu, value)
    else:
        return int(user_n)

def check_minus(text_menu):
    print(text_menu)
    user_n = input()
    if not user_n.replace('.', '', 1).isdigit() or user_n.replace('.', '', 1).isdigit() and
float(user_n) <= 0:
        return check_float(text_menu)
    else:
        return float(user_n)

def check_float(text_menu):
    print(text_menu)
    user_n = input()
    if not user_n.replace('.', '', 1).isdigit():
        return check_float(text_menu)
    else:
        return float(user_n)

def podstanovka(spisok, expression, spisok_letters):
    i = 0
    while i < len(spisok_letters):
        letter = spisok_letters[i]
        value = spisok[i]
        expression = expression.replace(letter, str(value))
        i += 1
    # print(expression)
    x = bool(eval(expression))
    print(x)
    print()

def standard_calc():
    print('Используется стандартный калькулятор.')
    print('Введите выражение:', end='')
    x = input()
    y = eval(x)
    print(round(y, 2))

def mass_calc():
    mas = [['1', '0.2', '0.0002', '200', '0.007054792389916097', '0.00044092452436976',
'2*10**(-7)'],
['5', '1', '0.001', '1000', '0.03527396194958048', '0.0022046226218488',
'0.000001'],
['5000', '1000', '1', '1000000', '35.27396194958048', '2.2046226218488',
'0.001'],
['0.005', '0.001', '0.000001', '1', '0.00003527396194958048',
'0.0000022046226218488', '10**(-9)'],
['141.74761562499972', '28.349523124999944', '0.028349523124999946',
'28349.523124999945', '1',
'0.062500000000000057', '0.000028349523124999945']]

```

```

        ['2267.961849999975', '453.592369999995', '0.453592369999995',
'453592.369999995', '15.99999999999854',
'1', '0.000453592369999995'],
        ['5000000', '1000000', '1000', '1000000000', '35273.96194958048',
'2204.6226218488', '1']]
    print('Используется калькулятор масс.')
    text_menu = ('Выберите единицу веса из которой нужно перевести:\n'
        '1) Карат\n'
        '2) Грамм\n'
        '3) Киллограмм\n'
        '4) Миллиграмм\n'
        '5) Унция\n'
        '6) Фунт\n'
        '7) Тонна')
    number_of_val1 = check(text_menu, 7)

    text_menu = ('Выберите единицу веса в которую нужно перевести:\n'
        '1) Карат\n'
        '2) Грамм\n'
        '3) Киллограмм\n'
        '4) Миллиграмм\n'
        '5) Унция\n'
        '6) Фунт\n'
        '7) Тонна')
    number_of_val2 = check(text_menu, 7)
    text_menu = 'Введите количество единиц массы'
    number = check_float(text_menu)
    y = str(number) + '*' + mas[number_of_val1 - 1][number_of_val2 - 1]
    x = eval(y)
    print(round(x, 5))

```

```

def currency_calc():
    import requests
    print('Используется калькулятор валют.')
    text_menu = ('Выберите валюту из которой переводите:\n'
        '1) RUB\n'
        '2) USD\n'
        '3) EUR\n'
        '4) GBP (Фунты)\n'
        '5) JPY (Японская йена)\n'
        '6) UAH (Гривны)')
    from_ = check(text_menu, 6)
    text_menu = ('Выберите валюту в которую надо перевести:\n'
        '1) RUB\n'
        '2) USD\n'
        '3) EUR\n'
        '4) GBP (Фунты)\n'
        '5) JPY (Японская йена)\n'
        '6) UAH (Гривны)')
    to_ = check(text_menu, 6)

    text_menu = 'Введите количество валюты'
    number = check_float(text_menu)

    if from_ == 1:
        url = 'https://v6.exchangerate-api.com/v6/5fe932e16be392834a80ef8f/latest/RUB'
        response = requests.get(url)
        data = response.json()
        if to_ == 1:
            y = str(number) + '*' + str(data['conversion_rates']['RUB'])
            x = eval(y)
            print(x)
        elif to_ == 2:
            y = str(number) + '*' + str(data['conversion_rates']['USD'])
            x = eval(y)
            print(x)
        elif to_ == 3:
            y = str(number) + '*' + str(data['conversion_rates']['EUR'])
            x = eval(y)
            print(x)
        elif to_ == 4:
            y = str(number) + '*' + str(data['conversion_rates']['GBP'])
            x = eval(y)
            print(x)
        elif to_ == 5:
            y = str(number) + '*' + str(data['conversion_rates']['JPY'])

```

```

        x = eval(y)
        print(x)
    elif to_ == 6:
        y = str(number) + '*' + str(data['conversion_rates']['UAH'])
        x = eval(y)
        print(x)
elif from_ == 2:
    url = 'https://v6.exchangerate-api.com/v6/5fe932e16be392834a80ef8f/latest/USD'
    response = requests.get(url)
    data = response.json()
    if to_ == 1:
        y = str(number) + '*' + str(data['conversion_rates']['RUB'])
        x = eval(y)
        print(x)
    elif to_ == 2:
        y = str(number) + '*' + str(data['conversion_rates']['USD'])
        x = eval(y)
        print(x)
    elif to_ == 3:
        y = str(number) + '*' + str(data['conversion_rates']['EUR'])
        x = eval(y)
        print(x)
    elif to_ == 4:
        y = str(number) + '*' + str(data['conversion_rates']['GBP'])
        x = eval(y)
        print(x)
    elif to_ == 5:
        y = str(number) + '*' + str(data['conversion_rates']['JPY'])
        x = eval(y)
        print(x)
    elif to_ == 6:
        y = str(number) + '*' + str(data['conversion_rates']['UAH'])
        x = eval(y)
        print(x)
elif from_ == 3:
    url = 'https://v6.exchangerate-api.com/v6/5fe932e16be392834a80ef8f/latest/EUR'
    response = requests.get(url)
    data = response.json()
    if to_ == 1:
        y = str(number) + '*' + str(data['conversion_rates']['RUB'])
        x = eval(y)
        print(x)
    elif to_ == 2:
        y = str(number) + '*' + str(data['conversion_rates']['USD'])
        x = eval(y)
        print(x)
    elif to_ == 3:
        y = str(number) + '*' + str(data['conversion_rates']['EUR'])
        x = eval(y)
        print(x)
    elif to_ == 4:
        y = str(number) + '*' + str(data['conversion_rates']['GBP'])
        x = eval(y)
        print(x)
    elif to_ == 5:
        y = str(number) + '*' + str(data['conversion_rates']['JPY'])
        x = eval(y)
        print(x)
    elif to_ == 6:
        y = str(number) + '*' + str(data['conversion_rates']['UAH'])
        x = eval(y)
        print(x)
elif from_ == 4:
    url = 'https://v6.exchangerate-api.com/v6/5fe932e16be392834a80ef8f/latest/GBP'
    response = requests.get(url)
    data = response.json()
    if to_ == 1:
        y = str(number) + '*' + str(data['conversion_rates']['RUB'])
        x = eval(y)
        print(x)
    elif to_ == 2:
        y = str(number) + '*' + str(data['conversion_rates']['USD'])
        x = eval(y)
        print(x)
    elif to_ == 3:
        y = str(number) + '*' + str(data['conversion_rates']['EUR'])
        x = eval(y)

```

```

        print(x)
    elif to_ == 4:
        y = str(number) + '*' + str(data['conversion_rates']['GBP'])
        x = eval(y)
        print(x)
    elif to_ == 5:
        y = str(number) + '*' + str(data['conversion_rates']['JPY'])
        x = eval(y)
        print(x)
    elif to_ == 6:
        y = str(number) + '*' + str(data['conversion_rates']['UAH'])
        x = eval(y)
        print(x)
elif from_ == 5:
    url = 'https://v6.exchangerate-api.com/v6/5fe932e16be392834a80ef8f/latest/JPY'
    response = requests.get(url)
    data = response.json()
    if to_ == 1:
        y = str(number) + '*' + str(data['conversion_rates']['RUB'])
        x = eval(y)
        print(x)
    elif to_ == 2:
        y = str(number) + '*' + str(data['conversion_rates']['USD'])
        x = eval(y)
        print(x)
    elif to_ == 3:
        y = str(number) + '*' + str(data['conversion_rates']['EUR'])
        x = eval(y)
        print(x)
    elif to_ == 4:
        y = str(number) + '*' + str(data['conversion_rates']['GBP'])
        x = eval(y)
        print(x)
    elif to_ == 5:
        y = str(number) + '*' + str(data['conversion_rates']['JPY'])
        x = eval(y)
        print(x)
    elif to_ == 6:
        y = str(number) + '*' + str(data['conversion_rates']['UAH'])
        x = eval(y)
        print(x)
elif from_ == 6:
    url = 'https://v6.exchangerate-api.com/v6/5fe932e16be392834a80ef8f/latest/UAH'
    response = requests.get(url)
    data = response.json()
    if to_ == 1:
        y = str(number) + '*' + str(data['conversion_rates']['RUB'])
        x = eval(y)
        print(x)
    elif to_ == 2:
        y = str(number) + '*' + str(data['conversion_rates']['USD'])
        x = eval(y)
        print(x)
    elif to_ == 3:
        y = str(number) + '*' + str(data['conversion_rates']['EUR'])
        x = eval(y)
        print(x)
    elif to_ == 4:
        y = str(number) + '*' + str(data['conversion_rates']['GBP'])
        x = eval(y)
        print(x)
    elif to_ == 5:
        y = str(number) + '*' + str(data['conversion_rates']['JPY'])
        x = eval(y)
        print(x)
    elif to_ == 6:
        y = str(number) + '*' + str(data['conversion_rates']['UAH'])
        x = eval(y)
        print(x)

def ipoteka_calc():
    print('Используется калькулятор ипотеки.')
    print('Процентная ставка банка 10%')
    text_menu = 'Стоимость недвижимости:'
    price = check_minus(text_menu)
    text_menu = 'Первоначальный взнос:'

```

```

first_in = check_minus(text_menu)
text_menu = 'Срок кредита в годах(целое число от 1 до 30 лет):'
years = check(text_menu, 30)

month_st = 10 / 12 / 100
ob_st = (1 + month_st) ** (12 * years)
month_pay = (price - first_in) * month_st * ob_st / (ob_st - 1)
overpay = month_pay * (years * 12) - (price - first_in)

print('Сумма ежемесячного платежа:', round(month_pay, 2))
print('Общая сумма переплаты:', round(overpay, 2))

# print('Сумма ежемесячного платежа:', round(((price - first_in) * 1.1 ** years) / (12 *
years), 2))
# print('Сумма общей переплаты:', round(((price - first_in) * 1.1 ** years) - (price -
first_in), 2))

def temp_calc():
    print('Используется калькулятор температуры.')
    mas = [['1', '33.8', '274.15', '493.47', '0.8'],
            ['0.02958579881656805', '1', '8.11094674556213', '14.599704142011836',
            '0.02366863905325444'],
            ['0.0036476381542950944', '0.12329016961517418', '1', '1.8000000000000003',
            '0.0029181105234360755'],
            ['0.002026465641275052', '0.06849453867509675', '0.5555555555555555', '1',
            '0.0016211725130200417'],
            ['1.25', '42.24999999999999', '342.68749999999994', '616.8375', '1']]
    text_menu = ('Выберите единицу температуры из которой нужно перевести:\n'
                 '1) Цельсий\n'
                 '2) Фаренгейт\n'
                 '3) Кельвин\n'
                 '4) Рэнкин\n'
                 '5) Réaumur')
    number_of_val1 = check(text_menu, 5)
    text_menu = ('Выберите единицу температуры в которую нужно перевести:\n'
                 '1) Цельсий\n'
                 '2) Фаренгейт\n'
                 '3) Кельвин\n'
                 '4) Рэнкин\n'
                 '5) Réaumur')
    number_of_val2 = check(text_menu, 5)
    text_menu = 'Введите количество единиц температуры:'
    number = check_float(text_menu)
    y = str(number) + '*' + mas[number_of_val1 - 1][number_of_val2 - 1]
    x = eval(y)
    print(round(x, 3))

def log_calc():
    print('Используется логический калькулятор.')
    print('Обозначения символов:\n'
          '\& : * (логическое И)\n'
          '\& : + (логическое ИЛИ)\n'
          '=' : '=' (логическое тождество)\n'
          '!' : '!' (логическое отрицание)\n'
          '-> : '<=' (логическое следование)\n")

    print('Введите логическое выражение используя предложенные обозначения')
    print('Если вы что-либо сравниваете операцией "==" , то вам необходимо поставить скобки
    вокруг обоих сравниваемых выражений')
    print('Для корректной работы программы после каждого введенного символа стоит ставить
    пробел!\n')
    log_ = input()
    temp = log_

    # temp = 'A * (A + B) == !C'
    temp = temp.replace('*', ' & ')
    temp = temp.replace('+', ' & ')
    temp = temp.replace('==', '=')
    temp = temp.replace('(', ' ( ')
    temp = temp.replace(')', ' ) ')
    temp = temp.replace('!', ' ! ')
    temp = temp.replace('<=', '<= ')
    mas = temp.split(' ')
    i = 0
    while i < len(mas):

```

```

        k = 0
        while k < len(mas):
            if mas[i] == mas[k] and i != k:
                mas.pop(k)
                k += 1
            i += 1
        # print(mas)
        i = 0
        while i < len(mas):
            if mas[i] == '':
                mas.pop(i)
                i -= 1
            i += 1

        print(mas)
        print()

        log_ = log_.replace('*', 'and')
        log_ = log_.replace('+', 'or')
        log_ = log_.replace('!', 'not ')
        # log_.replace('++', '')

        list_ = []
        for i in range(len(mas)):
            list_.append(0)
        print(list_)

        podstanovka(list_, log_, mas)
        j = len(list_) - 1
        while j >= 0:
            if list_[j] == 0:
                list_[j] = 1
                k = j + 1
                while k < len(list_):
                    list_[k] = 0
                    k += 1
                print(list_)
                podstanovka(list_, log_, mas)
                j = len(list_) - 1
            else:
                j -= 1
        """if 0 or (0 and 0) == not 1:
            print('yes')"""

number_of_calc = main_menu()
while True:
    if number_of_calc != '1' and number_of_calc != '2' and number_of_calc != '3' and
number_of_calc != '4' and number_of_calc != '5' and number_of_calc != '6':
        number_of_calc = main_menu()
    else:
        if number_of_calc == '1':
            standard_calc()
        elif number_of_calc == '2':
            mass_calc()
        elif number_of_calc == '3':
            currency_calc()
        elif number_of_calc == '4':
            ipoteka_calc()
        elif number_of_calc == '5':
            temp_calc()
        elif number_of_calc == '6':
            log_calc()
        break

```