# Архітектура комп’ютера

# Індивідуальне Домашнє Завдання

# Програмна інженерія

# Група 6.1219-2

# Піддубняк Данііл

# Програма «Калькулятор»

# Приклад роботи(Скріншоти):

# Код програми:

import time  
"""Описание:  
Калькулятор выполняет обычные арифметические операция над двумя вводимыми числами в произвольных СС  
Ответ можно получить в произвольной СС  
Операции производятся над отрицательными и дробными числами в том числе.  
Выполнил:  
Поддубняк Д.  
Группа 6.1219-2  
Приятного пользования!"""  
  
def literator(Num): # Перевод цифр в буквенные соответствия СС.  
 Num = list(Num.upper()) # Число переводится в список(упрощенный вывод)  
 while "A" in Num:  
 NumIndex = Num.index('A')  
 Num[NumIndex] = '10'  
  
 while "B" in Num:  
 NumIndex = Num.index('B')  
 Num[NumIndex] = '11'  
  
 while "C" in Num:  
 NumIndex = Num.index('C')  
 Num[NumIndex] = '12'  
  
 while "D" in Num:  
 NumIndex = Num.index('D')  
 Num[NumIndex] = '13'  
  
 while "E" in Num:  
 NumIndex = Num.index('E')  
 Num[NumIndex] = '14'  
  
 while "F" in Num:  
 NumIndex = Num.index('F')  
 Num[NumIndex] = '15'  
 return (Num)  
  
  
def Solution(SystemNum, OutputNum, Accuracy, Num):  
 Num = str(Num)  
 NumRow = '0123456789ABCDEF' # Алфавит(ограничение перевода до 16 СС.)  
 try:  
 dot = Num.index('.') - 1 # Получение индекса точки для вычислений  
 k = literator(Num) # Коррекция цифр в буквенные индексы  
 item = '.'  
 if item in k:  
 k.remove(item)  
 k = list(map(str, k))  
 k = list(map(int, k))  
 # Исключение постороних символов (!"№;%:?\*()" и.т.д)  
 except ValueError:  
 return "Неверно введено число."  
 else:  
  
 # Перевод в 10-ю систему СС для дальнешего перевода  
 Res = 0  
  
 for elem in k:  
 Res += int(elem) \* (SystemNum \*\* dot)  
 dot -= 1  
 Res = str(Res)  
  
 # Отделение целой части от дробной  
 firstdigit, seconddigit = Res.split('.')  
 firstdigit = int(firstdigit) # Целая часть  
 seconddigit = int(seconddigit) / 10 \*\* (len(seconddigit)) # Дробная часть  
  
 # Используется для упрощенного совмещения частей числа  
 IntPartList = [] # Список целой части числа  
 FracPartList = [] # Список дробной части числа  
 # Перевод целой части  
 while firstdigit != 0:  
 IntPartList.append(NumRow[firstdigit % OutputNum]) # Ввод в список целой части  
 firstdigit = firstdigit // OutputNum  
 # #Перевод дробной части  
 for j in range(Accuracy):  
 for i in str(seconddigit):  
 seconddigit = seconddigit \* OutputNum  
 if seconddigit < 1:  
 FracPartList.append(NumRow[int(seconddigit)]) # Ввод в список дробной части  
 else:  
 FracPartList.append(NumRow[int(seconddigit)])  
 sdstr = str(seconddigit)  
 sdstr = sdstr.split('.')  
 seconddigit = int(sdstr[1]) / 10 \*\* (len(sdstr[1]))  
 return ("".join(map(str, IntPartList[::-1])) + '.' + "".join(map(str, FracPartList[:Accuracy:1]))) # Возврат итога производится в строке  
  
  
  
def continued(num):  
 while True:  
 print('=' \* 50)  
 mode = input('''Выберите действие:  
1.Продолжить вычисления.  
2.Новое число.  
3.Выход.\n''')  
 print('=' \* 50)  
 if mode == '1':  
 todo = input("Введите действие(+, -, \*, /):\n")  
 second\_system = int(input("Введите систему счисления второго числа:\n"))  
 if second\_system <= 16 and second\_system >= 2: # Ограничение СС  
 second\_digit = input("Введите второе число:\n")  
 if '.' in second\_digit: # Если число дробное  
 if '-' in second\_digit: # Если отрицательное  
 split = second\_digit.split('-', 2) # Отделение минуса от числа  
 split[0] = '-'  
 second\_digit = split[1] # Число отделено от минуса  
 second\_d = float(split[0] + Solution(second\_system, 10, 2, second\_digit)) # Перевод числа  
 else:  
 second\_d = float(Solution(second\_system, 10, 2, second\_digit)) # Перевод числа  
 else:  
 # Перевод целочисленно если число не дробное  
 second\_d = int(convertor(second\_digit, 10, second\_system))  
 try:  
 # Выбор действия между числами  
 if todo == "+":  
 # Сложение  
 res = addition(num, second\_d)  
 elif todo == "-":  
 # Вычитание  
 res = subtraction(num, second\_d)  
 elif todo == "\*":  
 # Умножение  
 res = multiplication(num, second\_d)  
 elif todo == "/":  
 # Деление  
 res = division(num, second\_d)  
 else:  
 print("Неверно выбрано действие.")  
 break  
 try:  
 ss = int(input("Введите систему счисления для ответа:\n"))  
 if ss <= 16 and ss >= 2:  
 new\_res = str(res)  
 if '.' in new\_res: # Если дробное  
 if '-' in new\_res: # Если отрицательное  
 split = new\_res.split('-',2) # Отделение минуса  
 split[0] = '-'  
 new\_res = split[1] # Число без минуса  
 result = split[0] + Solution(10, ss, 2, new\_res) # Соединение минуса с переводом  
 else:  
 result = Solution(10, ss, 2, new\_res) # Перевод числа  
 else:  
 result = convertor(new\_res, ss) # Целочисленный перевод если не дробное  
 print('=' \* 50)  
 print(f"Ответ: {result}")  
 print('=' \* 50)  
 time.sleep(2) # Задержка вывода  
 continued(res) # Рекурсия функции-продолжения  
 else:  
 print("Неверно введена СС.Макс-16,Мин-2.")  
 break  
 except Exception:  
 print("СС от 2 до 16")  
 break  
 except Exception:  
 print("Ошибка")  
 break  
 else:  
 print("Неверно введена СС.Макс-16,Мин-2.")  
 break  
 elif mode == '2':  
 break # Окончание работы  
 else:  
 exit() # Окончание работы  
  
def start():  
 first\_system = int(input("Введите систему счисления вашего числа:\n"))  
 if first\_system <= 16 and first\_system >= 2: # Ограничение СС  
 first\_digit = input('Введите первое число:\n')  
 if '.' in first\_digit: # Если дробное  
 if '-' in first\_digit: # Если отрицательное  
 split = first\_digit.split('-', 2) # Отделение минуса от числа  
 split[0] = '-'  
 first\_digit = split[1] # Число без минуса  
 first\_d = float(split[0] + Solution(first\_system, 10, 2, first\_digit)) # Перевод числа  
 else:  
 first\_d = float(Solution(first\_system, 10, 2, first\_digit)) # Перевод числа  
 else:  
 first\_d = int(convertor(first\_digit, 10, first\_system)) # Целочисленный перевод если не дробное  
 todo = input("Введите действие(+, -, \*, /):\n") # Выбор действия  
 second\_system = int(input("Введите систему счисления второго числа:\n"))  
 if second\_system <= 16 and second\_system >= 2: # Ограничение СС  
 second\_digit = input("Введите второе число:\n")  
 if '.' in second\_digit: # Если дробное  
 if '-' in second\_digit: # Если отрицательное  
 split = second\_digit.split('-', 2) # Отделение минуса от числа  
 split[0] = '-'  
 second\_digit = split[1] # Число без минуса  
 second\_d = float(split[0] + Solution(second\_system, 10, 2, second\_digit)) # Перевод числа  
 else:  
 second\_d = float(Solution(second\_system, 10, 2, second\_digit)) # Перевод числа  
 else:  
 second\_d = int(convertor(second\_digit, 10, second\_system)) # Целочисленный перевод если не дробное  
 try:  
 if todo == "+":  
 res = addition(first\_d, second\_d)  
 elif todo == "-":  
 res = subtraction(first\_d, second\_d)  
 elif todo == "\*":  
 res = multiplication(first\_d, second\_d)  
 elif todo == "/":  
 res = division(first\_d, second\_d)  
 else:  
 print("Неверно выбрано действие.")  
 try:  
 ss = int(input("Введите систему счисления для ответа:\n"))  
 if ss <= 16 and ss >= 2:  
 new\_res = str(res)  
 if '.' in new\_res: # Если дробное  
 if '-' in new\_res: # Если отрицательное  
 split = new\_res.split('-', 2) # Отделение минуса  
 split[0] = '-'  
 new\_res = split[1] # Число без минуса  
 result = split[0] + Solution(10, ss, 2, new\_res) # Соединение минуса с переводом  
 else:  
 result = Solution(10, ss, 2, new\_res) # Перевод числа  
 else:  
 result = convertor(new\_res, ss) # Целочисленный перевод если не дробное  
 print('='\*50)  
 print(f"Ответ: {result}")  
 print('=' \* 50)  
 time.sleep(2)  
 continued(res)  
 else:  
 print("Неверно введена СС.Макс-16,Мин-2.")  
 except Exception:  
 print("Неверно введена СС.Макс-16,Мин-2.")  
 except Exception:  
 print("Ошибка.")  
 else:  
 print("Неверно введена СС.Макс-16,Мин-2.")  
 else:  
 print("Неверно введена СС.Макс-16,Мин-2.")  
  
  
  
def convertor(num, to\_base=10, from\_base=10):  
 *"""Функия целочисленного перевода"""* try:  
 num = str(num)  
 if '-' in num: # Если отрицательное  
 split = num.split('-',2) # Отделение минуса  
 split[0] = '-'  
 num = split[1] # Действия с числом без минуса  
 if isinstance(num, str):  
 n = int(num, from\_base)  
 else:  
 n = int(num)  
 alphabet = "0123456789ABCDEF"  
 if n < to\_base:  
 return split[0] + alphabet[n] # Возврат итога с минусом  
 else:  
 return split[0] + convertor(n // to\_base, to\_base) + alphabet[n % to\_base] # Возврат итога с минусом  
 else:  
 if isinstance(num, str):  
 n = int(num, from\_base)  
 else:  
 n = int(num)  
 alphabet = "0123456789ABCDEF"  
 if n < to\_base:  
 return alphabet[n]  
 else:  
 return convertor(n // to\_base, to\_base) + alphabet[n % to\_base]  
 except(ValueError):  
 return "Ошибка совпадения числа с его СС."  
  
  
def addition(a, b):  
 c = a + b  
 return c  
  
  
def subtraction(a, b):  
 c = a - b  
 return c  
  
  
def multiplication(a, b):  
 c = a \* b  
 return c  
  
  
def division(a, b):  
 c = a / b  
 return c  
  
  
view = "=" \* 50  
print('''{0}  
Calculator by Poddubnjak D.  
Float digits may be rounded!  
{1}'''.format(view, view))  
  
while True:  
 print('=' \* 50)  
 print("Выберите действие:")  
 print("1.Начать работу.")  
 print("Exit-выход.")  
 print('=' \* 50)  
 starter = input('\n')  
 if starter == "Exit":  
 exit()  
 elif starter == '1':  
 start()