

Implementation operations

Problem Statement

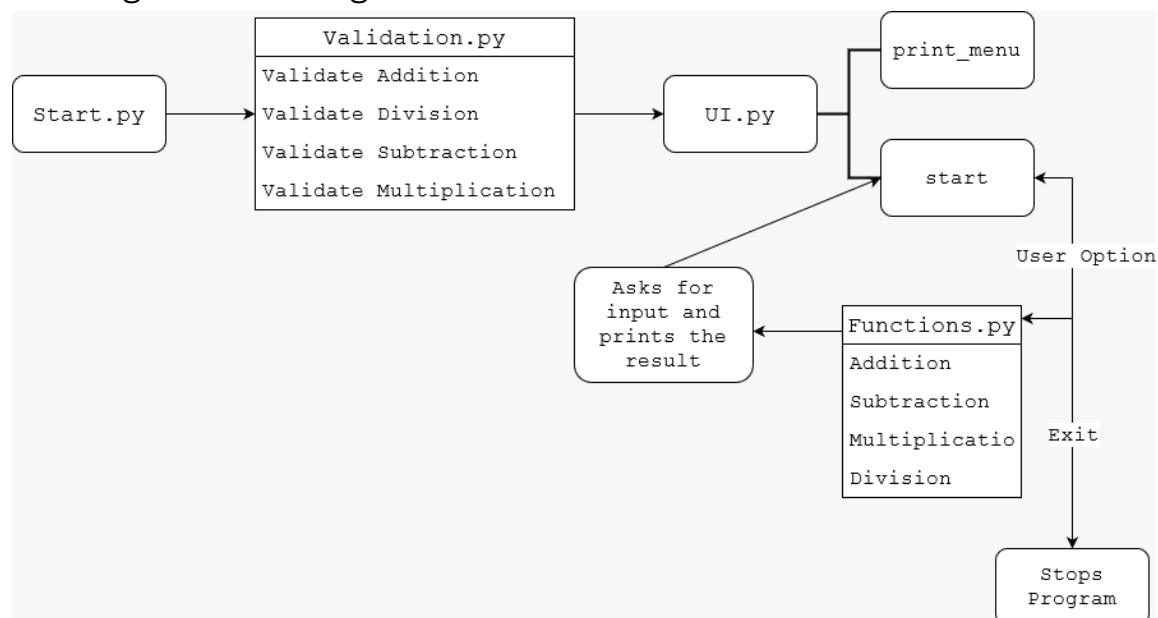
The program is designed to perform additions, subtractions, multiplications and divisions of numbers in a specified base, ranging from base 2, up to base 16.

The user will select the desired operation by choosing from the presented menu, he will enter the numbers he wishes to do a calculation on, choose the base of those numbers, and the program will display the computation and the result.

The program raises exceptions in case of invalid inputs, telling the user what caused the error.

A test module is also present, checking from the beginning, based on a given set of data, if the operations return the expected result.

Sub-Algorithm's Diagram



Pseudo-Code

ADDITION:

check_if_operation_is_valid(a, base)

check_if_operation_is_valid(b, base)

conversion_table = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 'A', 'B', 'C', 'D', 'E', 'F']

final_addition = ''

carry = 0

WHILE a OR b:

 last_digit_a = convert_str_to_int(a[LAST INDEX])

 last_digit_b = convert_str_to_int(b[LAST INDEX])

 addition = last_digit_a + last_digit_b + carry

 carry = conversion_table[addition // base]

 last_digit_addition = conversion_table[addition % base]

 final_addition = STRING(last_digit_addition) + final_addition

 REMOVE LAST CHARACTER FROM a

 REMOVE LAST CHARACTER FROM b

IF carry:

 final_addition = STRING(carry) + final_addition

WHILE FIRST CHARACTER OF final_addition EQUALS '0' AND LENGTH OF final_addition > 1:

 REMOVE FIRST CHARACTER FROM final_addition

RETURN final_addition

SUBTRACTION:

check_if_operation_is_valid(a, base)

check_if_operation_is_valid(b, base)

conversion_table = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 'A', 'B', 'C', 'D', 'E', 'F']

final_subtraction = ""

borrow = 0

IF LENGTH OF a < LENGTH OF b:

 SWAP a WITH b

WHILE a OR b:

 last_digit_a = convert_str_to_int(a[LAST INDEX])

 IF b:

 last_digit_b = convert_str_to_int(b[LAST INDEX])

 ELSE:

 last_digit_b = 0

 subtraction = last_digit_a - last_digit_b + borrow

 IF subtraction < 0:

 last_digit_subtraction = conversion_table[base + subtraction]

 borrow = -1

 ELSE:

 last_digit_subtraction = conversion_table[subtraction]

 borrow = 0

 final_subtraction = STRING(last_digit_subtraction) + final_subtraction

 REMOVE LAST CHARACTER FROM a

 REMOVE LAST CHARACTER FROM b

WHILE FIRST CHARACTER OF final_subtraction EQUALS '0' AND LENGTH OF final_subtraction > 1:

 REMOVE FIRST CHARACTER FROM final_subtraction

RETURN final_subtraction

MULTIPLICATION:

```
check_if_operation_is_valid(a, base)
```

```
check_if_operation_is_valid(b, base)
```

```
conversion_table = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 'A', 'B', 'C', 'D', 'E', 'F']
```

```
final_multiplication = ""
```

```
carry = 0
```

```
IF LENGTH OF a < LENGTH OF b:
```

```
    SWAP a WITH b
```

```
IF LENGTH OF b != 1:
```

```
    RAISE ValueError("Invalid numbers")
```

```
b = convert_str_to_int(b)
```

```
WHILE a:
```

```
    last_digit_a = convert_str_to_int(a[LENGTH(a)-1])
```

```
    multiplication = last_digit_a * b + convert_str_to_int(String(carry))
```

```
    carry = conversion_table[multiplication // base]
```

```
    last_digit_multiplication = conversion_table[multiplication % base]
```

```
    final_multiplication = String(last_digit_multiplication) + final_multiplication
```

```
    REMOVE LAST CHARACTER FROM a
```

```
IF carry:
```

```
    final_multiplication = String(carry) + final_multiplication
```

```
WHILE FIRST CHARACTER OF final_multiplication EQUALS '0' AND LENGTH OF final_multiplication > 1:
```

```
    REMOVE FIRST CHARACTER FROM final_multiplication
```

```
RETURN final_multiplication
```

DIVISION:

check_if_operation_is_valid(a, base)

check_if_operation_is_valid(b, base)

conversion_table = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 'A', 'B', 'C', 'D', 'E', 'F']

final_quotient = ""

IF LENGTH OF a < LENGTH OF b:

 SWAP a WITH b

IF LENGTH OF b != 1 OR b == '0':

 RAISE ValueError("Invalid numbers")

remainder = convert_str_to_int(a[FIRST INDEX])

b = convert_str_to_int(b)

WHILE a:

 quotient = conversion_table[remainder // b]

 remainder = conversion_table[remainder % b]

 final_quotient += STRING(quotient)

 REMOVE FIRST CHARACTER FROM a

 IF a IS EMPTY:

 BREAK

 remainder = convert_str_to_int(STRING(remainder)) * base + convert_str_to_int(a[FIRST INDEX])

WHILE FIRST CHARACTER OF final_quotient EQUALS '0' AND LENGTH OF final_quotient > 1:

 REMOVE FIRST CHARACTER FROM final_quotient

RETURN final_quotient, remainder