**OPERATIONS AND CONVERSIONS**

Moldovan Vasilica-Andreea

Group 915

1.Addition

The addition algorithm adds two given numbers in a given base and returns the result.

Pseudocode:

While length1 >= 0 do:

aux = digits[number1[length1]] + digits[number2[length2]] + carry

if aux >= base then:

aux -= base

carry = 1

else:  
 carry = 0

result += str(aux)

length1 -= 1

result = result[::-1]

return result

2.Subtraction

The subtraction algorithm subtracts two given numbers in a given base and returns the result.

Pseudocode:

While length1 >= 0 do:

aux = digits[number1[length1]] – digits[number2[length2]] + borrow

if aux < 0 the:

aux = aux + base

borrow = -1

else:

borrow = 0

result += str(aux)

result = result[::-1]

return result

3.Multiplication

The multiplication algorithm multiplies two given numbers in a certain base and returns the result.

Pseudocode:

For i := n; 0; -1:

Part\_sum = digits[i] \* digits[number2] + carry

Result = reverse\_digits[part\_sum%base] + result

Carry = part\_sum // base

If carry > 0 then:

Result = reverse\_digits[carry] + result

Return result

4.Division

The division algorithm divides two given natural numbers in a given base and returns the result.

Pseudocode:

I := 0

While I < length1 do:  
 aux += str(digits[number1[i]])

While aux // number2 == 0 do:

Aux += str(digits[number1[i]])

Carry = aux

Aux = aux // number2

Return aux

5.Conversions

5.1 Substitution method

This algorithm converts a number from a given base b into a given base h and returns the result, being recommended for b < h.

Pseudocode:

While n >= 0 do :

N = digits[number[i]]

Aux = multiplication(base2, p, n)

Nr = addition(base2, nr, aux)

P \*= base2

I -= 1

Return nr

Operations

Substitution method

Product

Addition

5.2 Successive divisions method

This algorithm converts a number from a given base b into a given base h and returns the result, being recommended for b > h.

Pseudocode:

While conversionToBaseTen(base1, number) >= conversionToBaseTen(base1, base2) do:

Aux = division(base1, number, base2)

Result = subtraction(base1, number, multiplication(base1, aux, base2))

Number = division(base1, number, base2)

Nr += str(conversionFromBaseTen(base2, result))

Nr += str(conversionToBaseTen(base1, number))

Nr = nr[::-1]

return nr

Successive divisions

Operation

Division

5.3 Conversions using an intermediate base (10)

This algorithm converts a number from a given base, base1 to a given base, base2, using base 10 as an intermediate base.

Pseudocode:

While n >= 0 do:

Aux = digits[number[i]]

Nr = nr + aux \* p

P \*= base

I -= 1

While number > 0 do:

Nr += digits[number % base]

Number = number // base

Nr = nr[::-1]

Return nr

Product

Addition

Operations

Intermediate base method

Main Diagram:

**Conversions**

**Operations**

**Main**