Exercitii Laborator

Structuri de date - Nicoleta Radu

int CmmdcEuclid\_1(int a, int b)

{

while (a!=b)

{

if (a > b)

{

a = a - b;

}

else

{

b = b - a;

}

}

return a;

}

// Function for CMMDC - Euclid - through division

void CmmdcEuclid\_2(int\* a, int\* b)

{

while (\*a > 1 && \*b > 1)

{

if (\*a > \*b)

{

\*a = \*a / \*b;

}

else

{

\*b = \*b / \*a;

}

}

}

// Modul

int Abs(int a)

{

if (a >= 0)

{

return a;

}

else

{

return a \* (-1);

}

}

// CMMDC

int cmmdcCiudatel(int a, int b)

{

int result{0};

int k{1};

while (a % 2 == 0 && b % 2 == 0)

{

a = a / 2;

b = b / 2;

k = 2 \* k;

}

int temp;

while (a > 0)

{

if (a % 2 == 0 || b % 2 == 0)

{

if (a % 2 == 0)

{

a = a / 2;

}

else

{

b = b / 2;

}

}

temp = (Abs(a - b) / 2);

if (a > b)

{

b = temp;

}

else

{

a = temp;

}

}

return k \* b;

}

// Exercitiu

/\*

std::vector<std::string> afterNDays(std::vector<std::string> days, int n)

{

std::vector<std::string> WeekDays = { "Monday", "Tuesday", "Wednesday", "Thursday","Friday", "Saturday", "Sunday"};

int index = 0;

int copy = n;

for (int i = 0; i < days.size(); i++)

{

copy = n;

for (int j = 0; j < WeekDays.size(); j++)

{

if (WeekDays[j] == days[i])

{

index = j;

if ((j + n) >= WeekDays.size())

{

if (n >= 7)

{

index = copy / 7;

cout << copy << " / " << 7 << " = " << index << endl;

index = index \* 7;

index = copy - index;

if (j + index >= WeekDays.size())

{

copy = ((WeekDays.size() - 1) - j);

index = (index - copy) - 1;

}

else

{

days[i] = WeekDays[index + j];

break;

}

}

else

{

index = ((WeekDays.size() - 1) - j);

index = (n - index) - 1;

}

days[i] = WeekDays[index];

break;

}

else

{

days[i] = WeekDays[index + n];

break;

}

}

}

}

return days;

}

\*/

// Ridicare la putere var 1

int Power(int num, int power)

{

int result = 1;

if (power == 0)

{

return 1;

}

if (power == 1)

{

return num;

}

while (power!=0)

{

result = result \* num;

power--;

}

return result;

}

int PowerVar2(int num, int power)

{

if (power == 0)

{

return 1;

}

if (power % 2 == 0)

{

return PowerVar2(num \* num, power / 2);

}

return num \* PowerVar2(num, power - 1);

}

// ITERATIV

int PowerVar3(int num, int power)

{

int result = 1;

if (power == 0)

{

return 1;

}

while (power != 0)

{

if (power % 2 == 0)

{

result = result \* num;

power = power / 2;

}

else

{

result = result \* num;

power = power - 1;

}

}

return result;

}