ALG. Fundamentali

# Slide 5, Problema 1

Sa se scrie un algoritm care preia ca data de intrare un sir X de numere naturale si returnweaza un sir care contine numarul de cifre pentru fieacre numar din sirul X

int getFigure(int n)

{

int count{ 0 };

while (n > 0)

{

n = n / 10;

count++;

}

return count;

}

int main()

{

vector <int> myVector{ 12,345,6545,23445,1,234 };

vector <int> mySecondVector(6, 0);

for (size\_t i = 0; i < myVector.size(); i++)

{

cout << myVector.at(i) << ", ";

}

cout << endl;

for (size\_t i = 0; i < myVector.size(); i++)

{

mySecondVector.at(i) = getFigure(myVector.at(i));

}

cout << endl;

for (size\_t i = 0; i < mySecondVector.size(); i++)

{

cout << mySecondVector.at(i) << ", ";

}

return 0;

}

# Slide 5 Problema 2

Afla daca nu numar este palindrom/superpalindrom

int isPalindrome(int a)

{

bool palindrome = true;

int copyN{ 0 };

int aux = a;

while (a > 0)

{

copyN = copyN \* 10 + a % 10;

a = a / 10;

}

if (copyN == aux)

{

palindrome = true;

}

else

{

palindrome = false;

}

return palindrome;

}

int isSuperPalindrome(int b)

{

bool superPalindrome = true;

int product = b \* b;

if (isPalindrome(b) == 1 && isPalindrome(product) == 1)

{

return superPalindrome;

}

else

{

return 0;

}

}

int main()

{

int number{ 1 };

cout << isSuperPalindrome(number);

return 0;

}

# Slide 6 problema 1