1. Check whether a number is a palindrome or not
2. Check whether an integer is prime or Not
3. Display prime numbers between two intervals
4. Check Armstrong number
5. Display Armstrong numbers between two intervals
6. Display factors of a number
7. Print pyramids and triangles
8. Create a simple calculator

Practice Exercises - Set I -Control Statements - 1-9-2020

Swaminathan Navinashok

2019115126

1.Check whether a number is a palindrome or not

#include<stdio.h>

int main()

{

int n,r,sum=0,temp;

printf("enter the number=");

scanf("%d",&n);

temp=n;

while(n>0)

{

r=n%10;

sum=(sum\*10)+r;

n=n/10;

}

if(temp==sum)

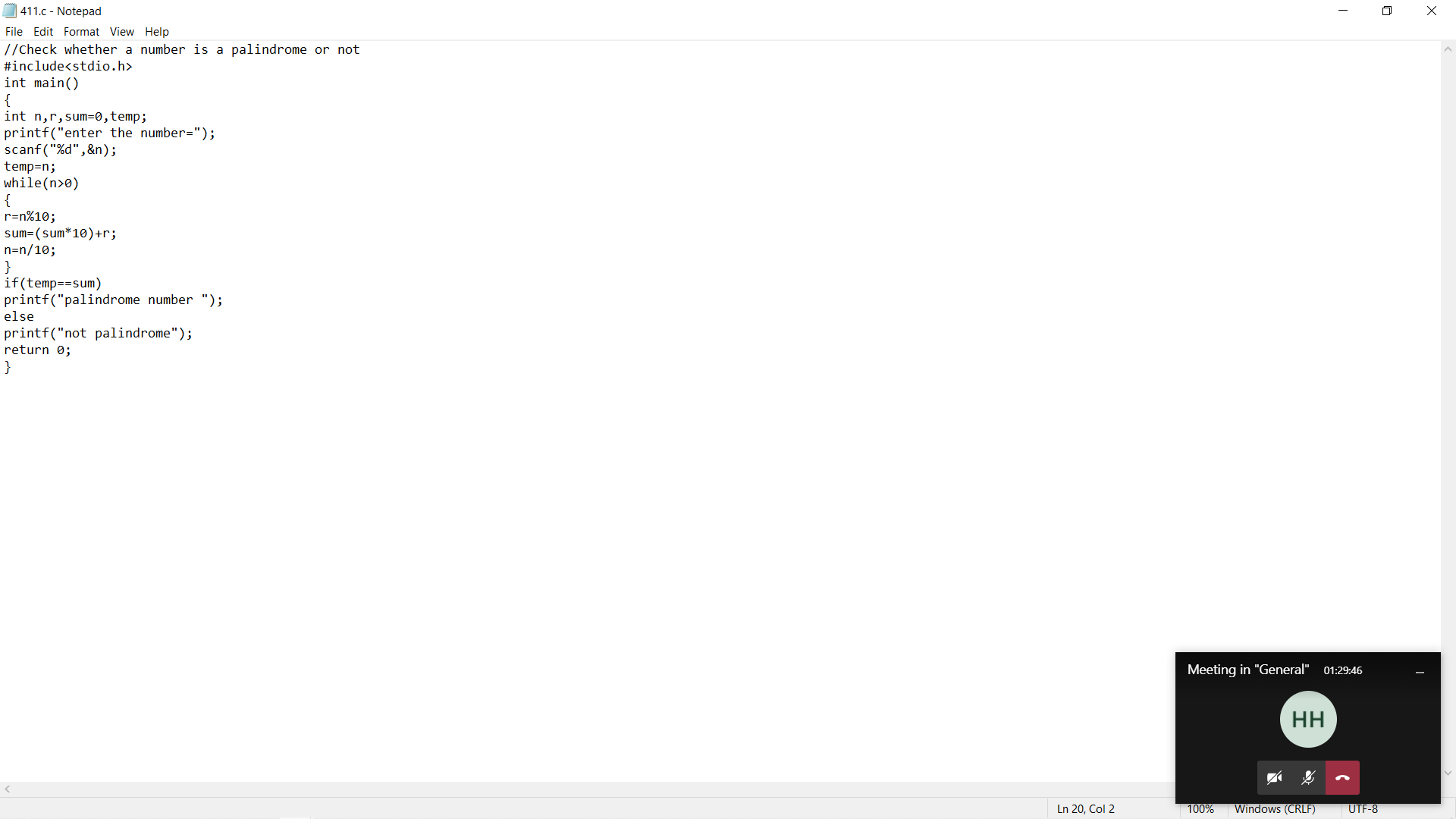
printf("palindrome number ");

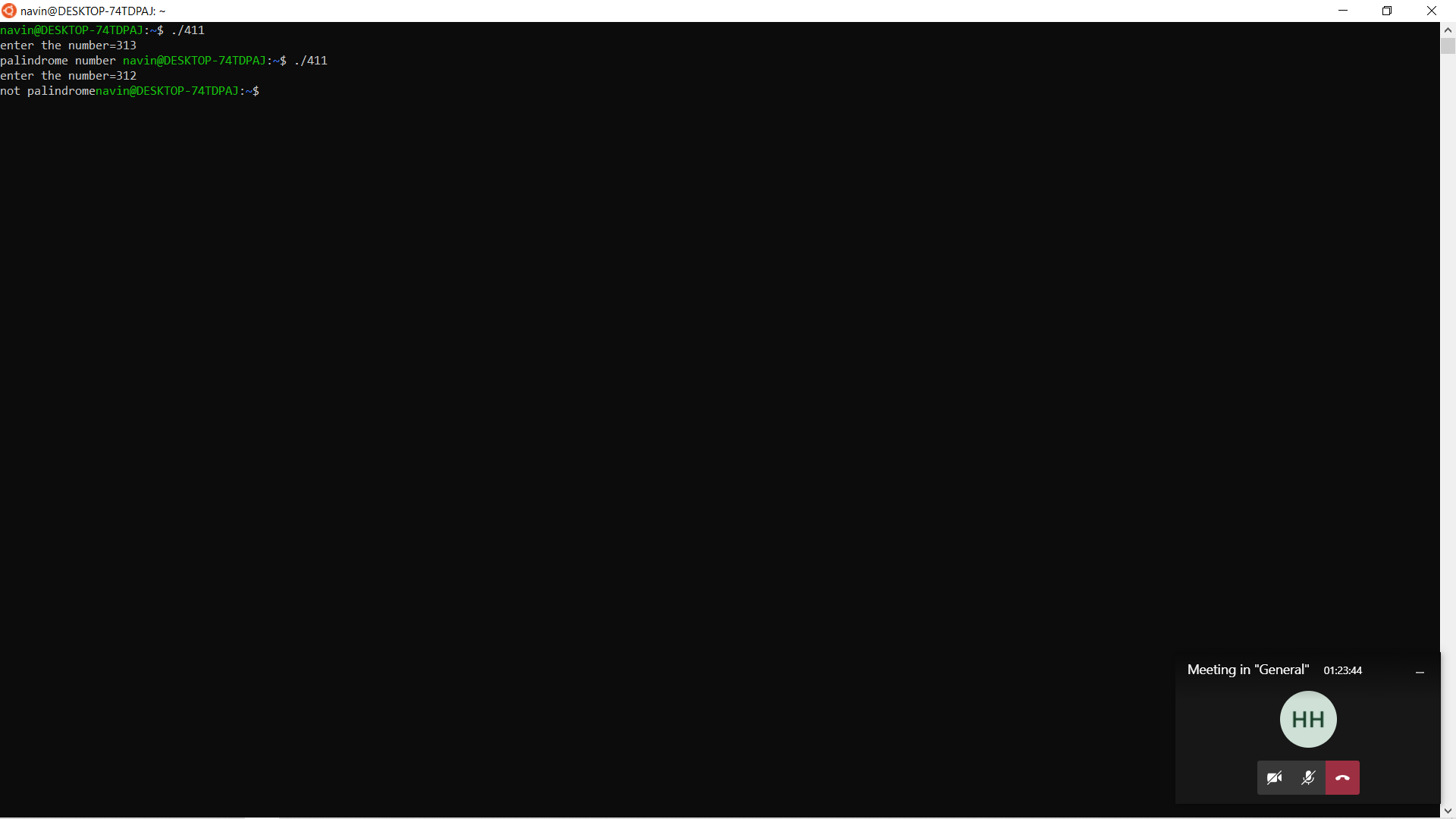
else

printf("not palindrome");

return 0;

}





2.Check whether an integer is prime or Not

#include <stdio.h>

int main() {

int n, i, flag = 0;

printf("Enter a positive integer: ");

scanf("%d", &n);

for (i = 2; i <= n / 2; ++i) {

// condition for non-prime

if (n % i == 0) {

flag = 1;

break;

}

}

if (n == 1) {

printf("1 is neither prime nor composite.");

}

else {

if (flag == 0)

printf("%d is a prime number.", n);

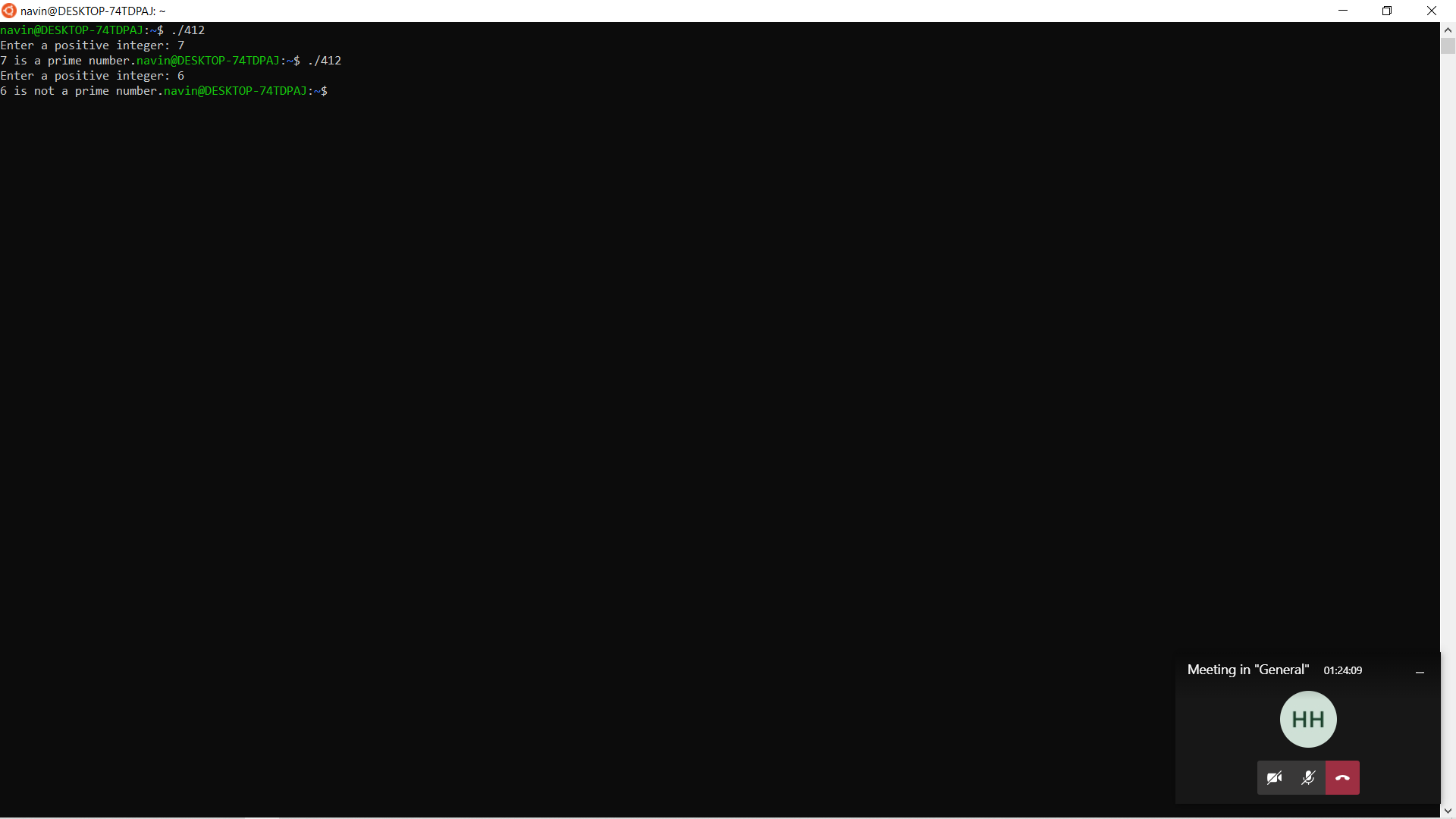
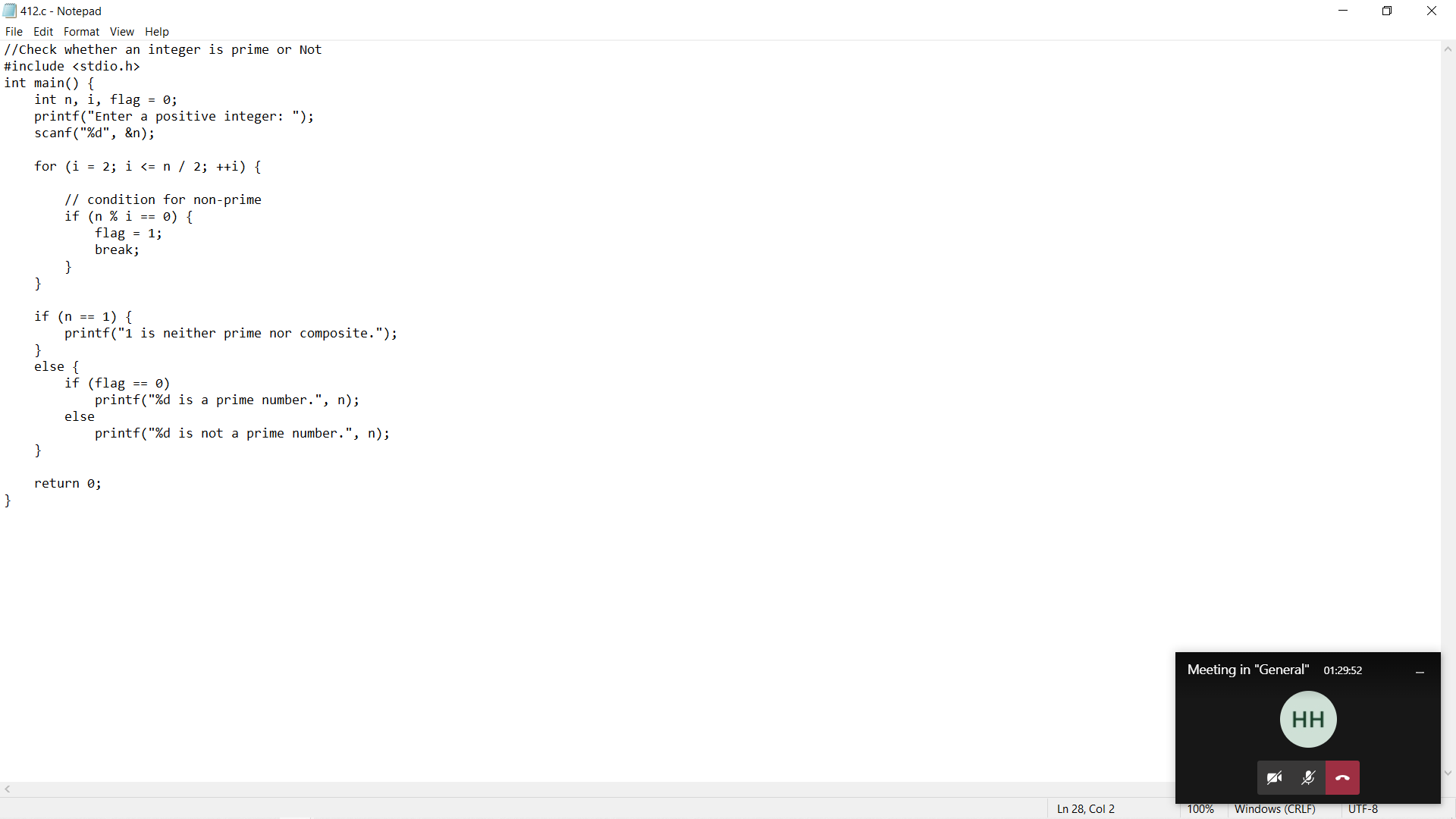
else

printf("%d is not a prime number.", n);

}

return 0;

}



3.Display prime numbers between two intervals

#include <stdio.h>

int main()

{

int low, high, i, flag;

printf("Enter two numbers(intervals): ");

scanf("%d %d", &low, &high);

printf("Prime numbers between %d and %d are: ", low, high);

// iteration until low is not equal to high

while (low < high) {

flag = 0;

// ignore numbers less than 2

if (low <= 1) {

++low;

continue;

}

// if low is a non-prime number, flag will be 1

for (i = 2; i <= low / 2; ++i) {

if (low % i == 0) {

flag = 1;

break;

}

}

if (flag == 0)

printf("%d ", low);

// to check prime for the next number

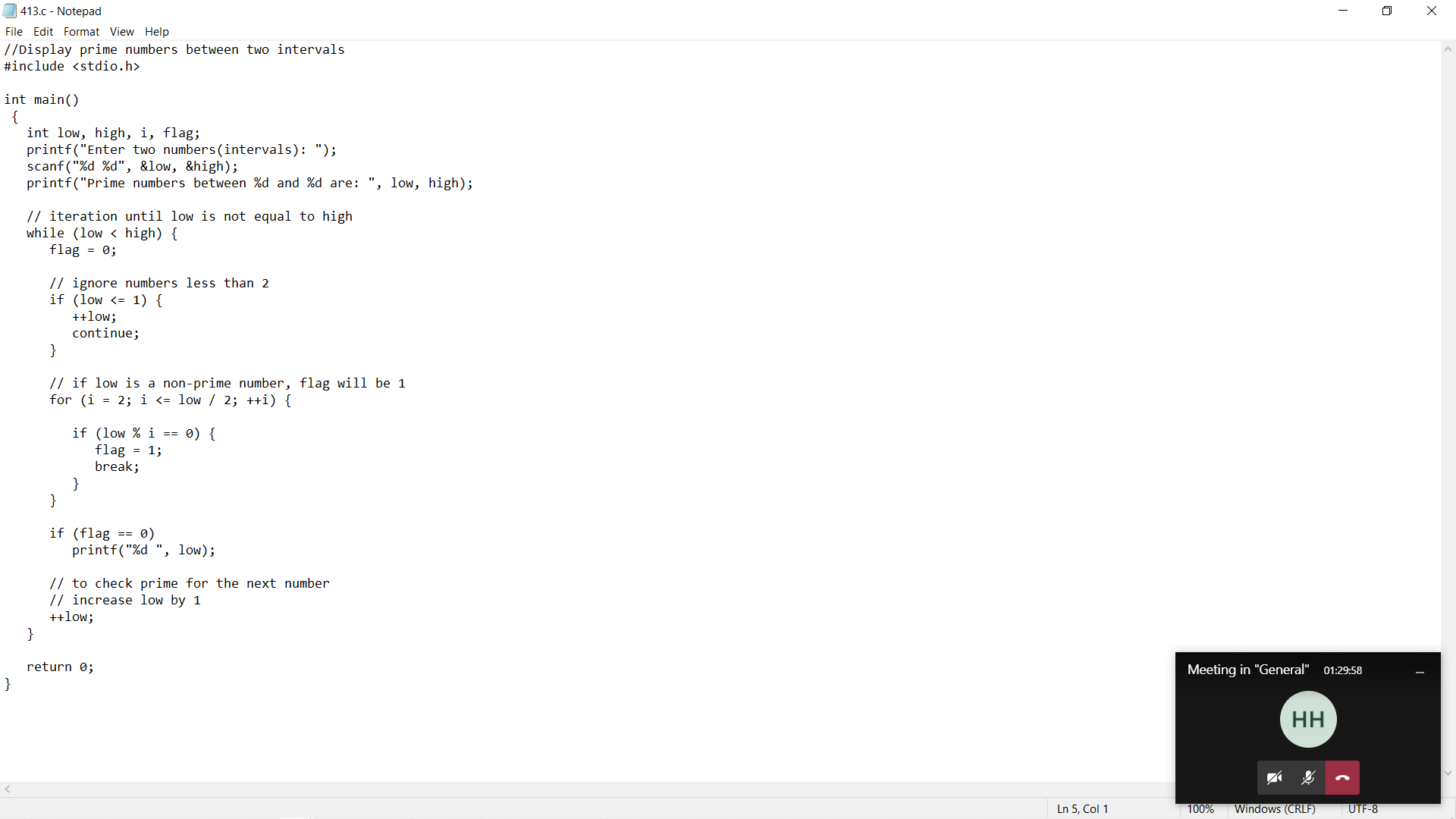
// increase low by 1

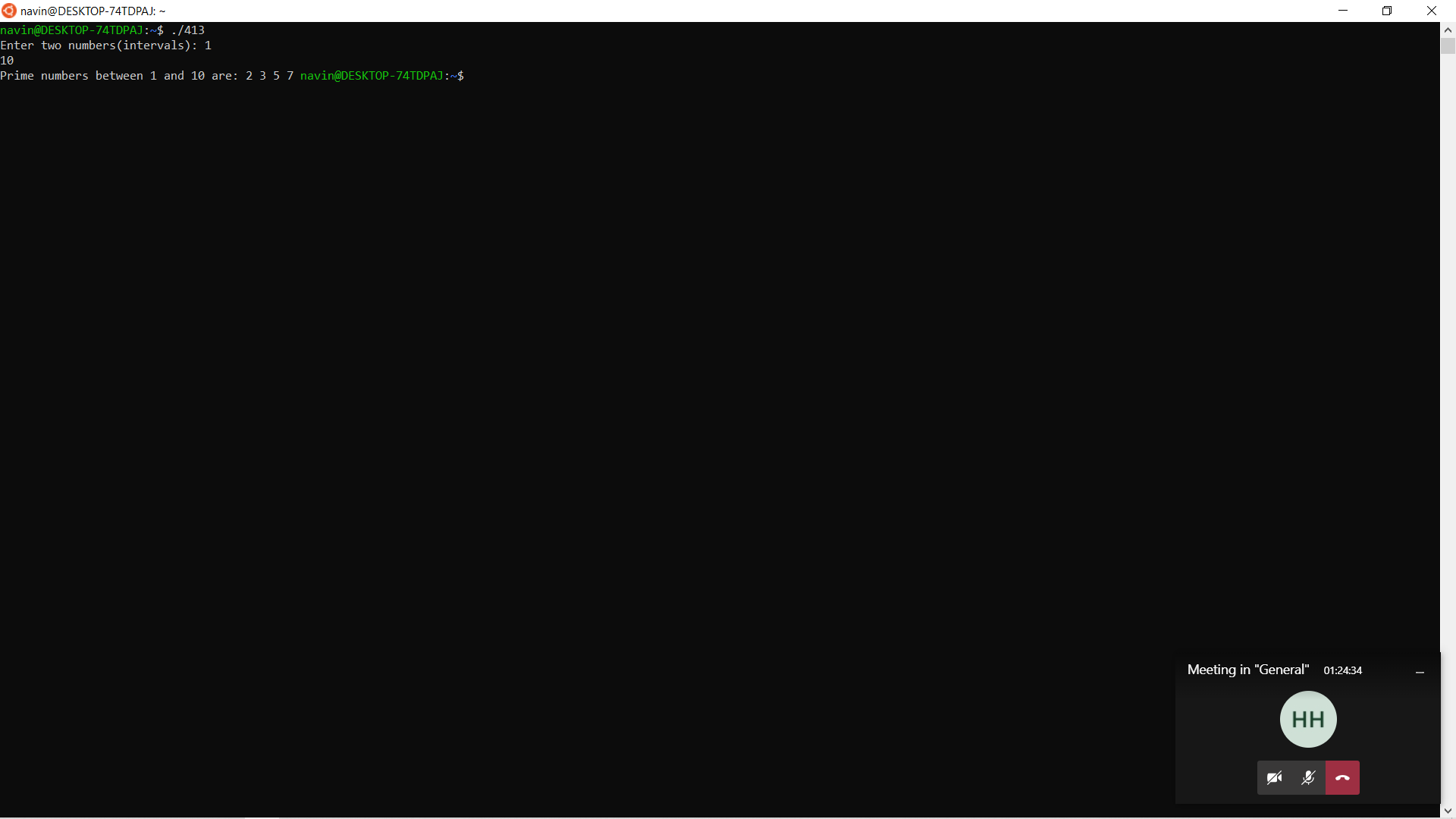
++low;

}

return 0;

}





4.Check Armstrong number

#include <stdio.h>

int main() {

int num, originalNum, remainder, result = 0;

printf("Enter a three-digit integer: ");

scanf("%d", &num);

originalNum = num;

while (originalNum != 0) {

// remainder contains the last digit

remainder = originalNum % 10;

result += remainder \* remainder \* remainder;

// removing last digit from the orignal number

originalNum /= 10;

}

if (result == num)

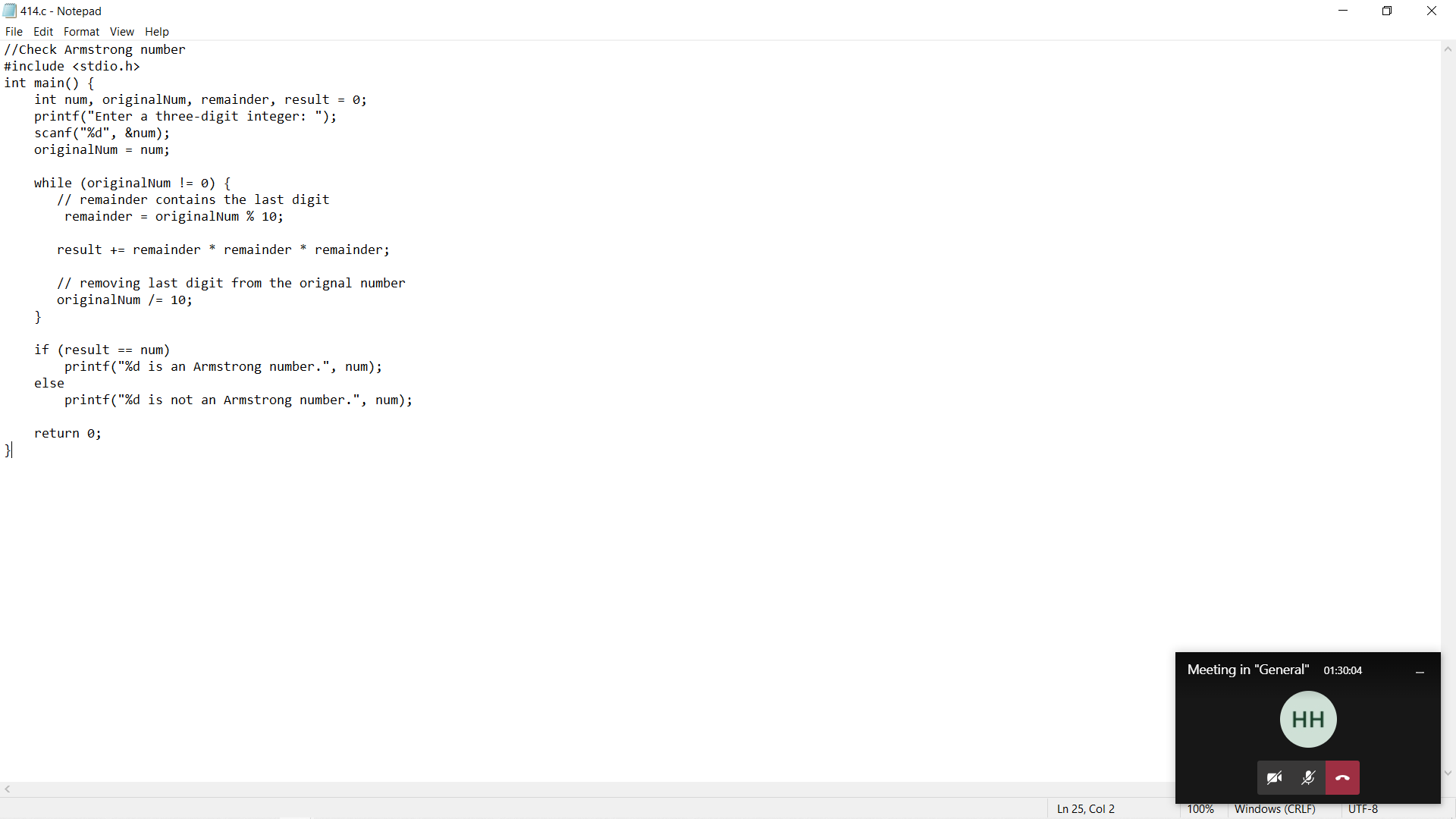
printf("%d is an Armstrong number.", num);

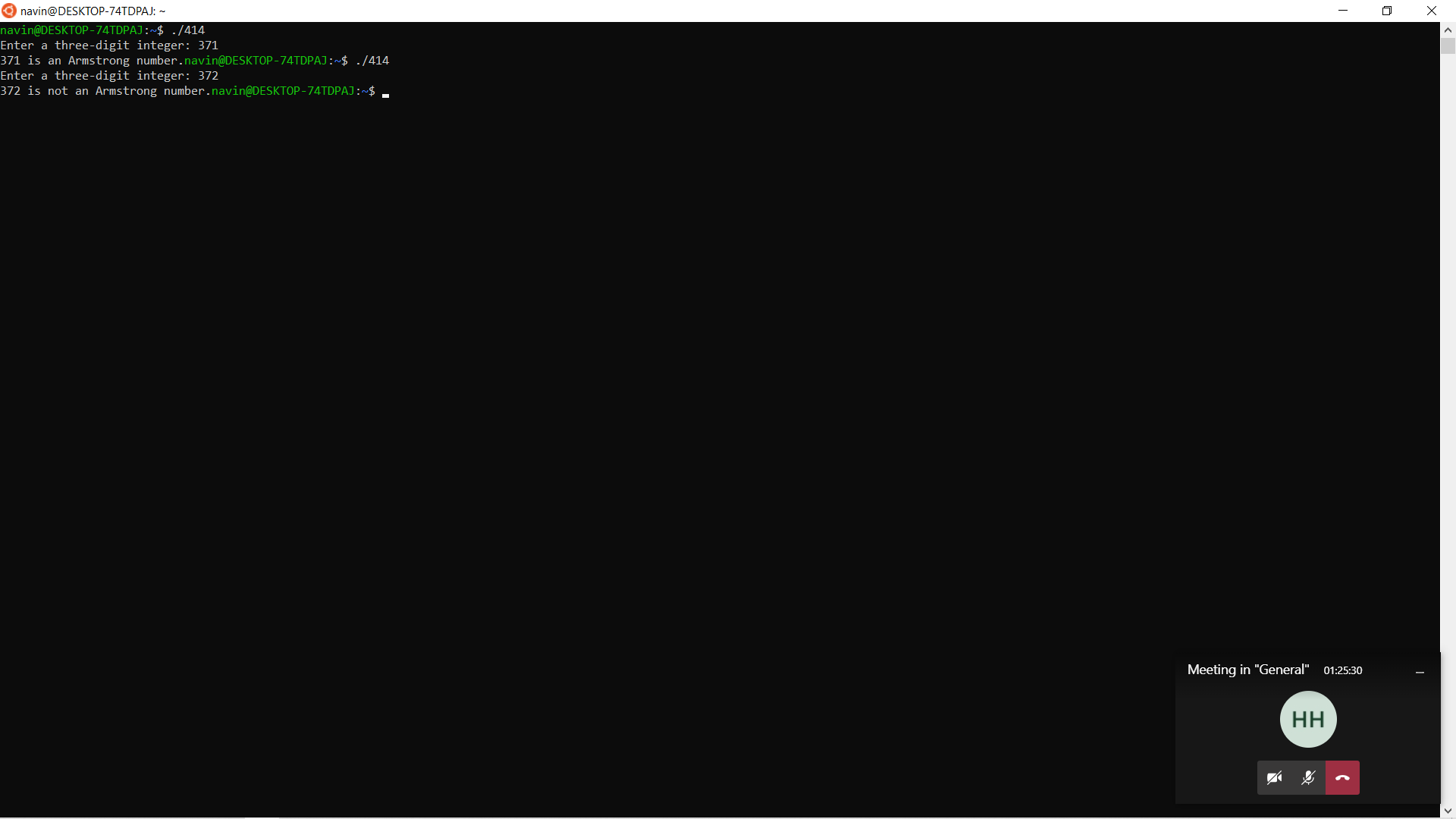
else

printf("%d is not an Armstrong number.", num);

return 0;

}





5.C program to print the Armstrong numbers between the two intervals

#include <stdio.h>

#include <math.h>

int main()

{

int start, end, i, temp1, temp2, remainder, n = 0, result = 0;

printf("Enter start value and end value : ");

scanf("%d %d", &start, &end);

printf("\nArmstrong numbers between %d an %d are: ", start, end);

for(i = start + 1; i < end; ++i)

{

temp2 = i;

temp1 = i;

while (temp1 != 0)

{

temp1 /= 10;

++n;

}

while (temp2 != 0)

{

remainder = temp2 % 10;

result += pow(remainder, n);

temp2 /= 10;

}

if (result == i) {

printf("%d ", i);

}

n = 0;

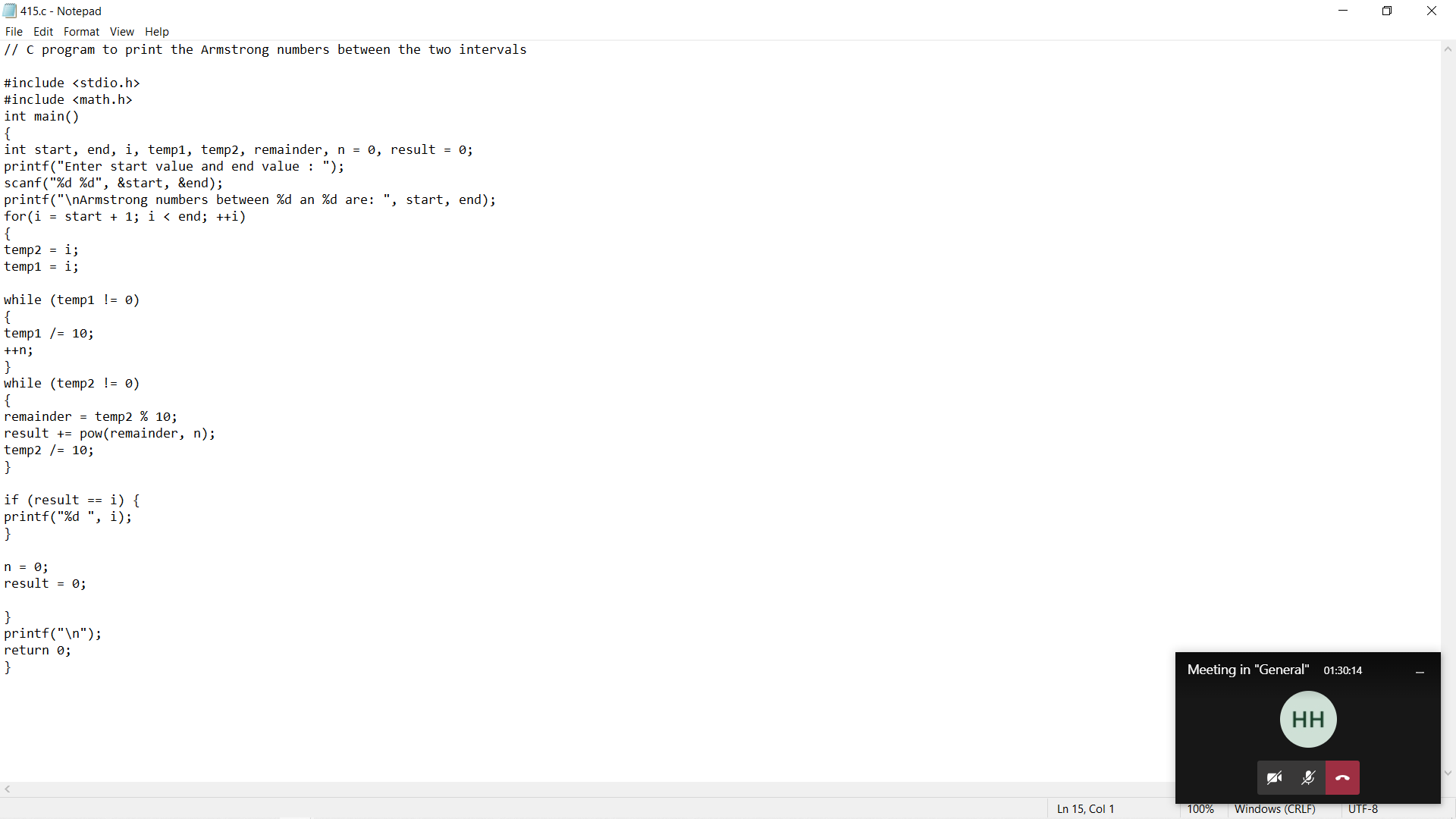
result = 0;

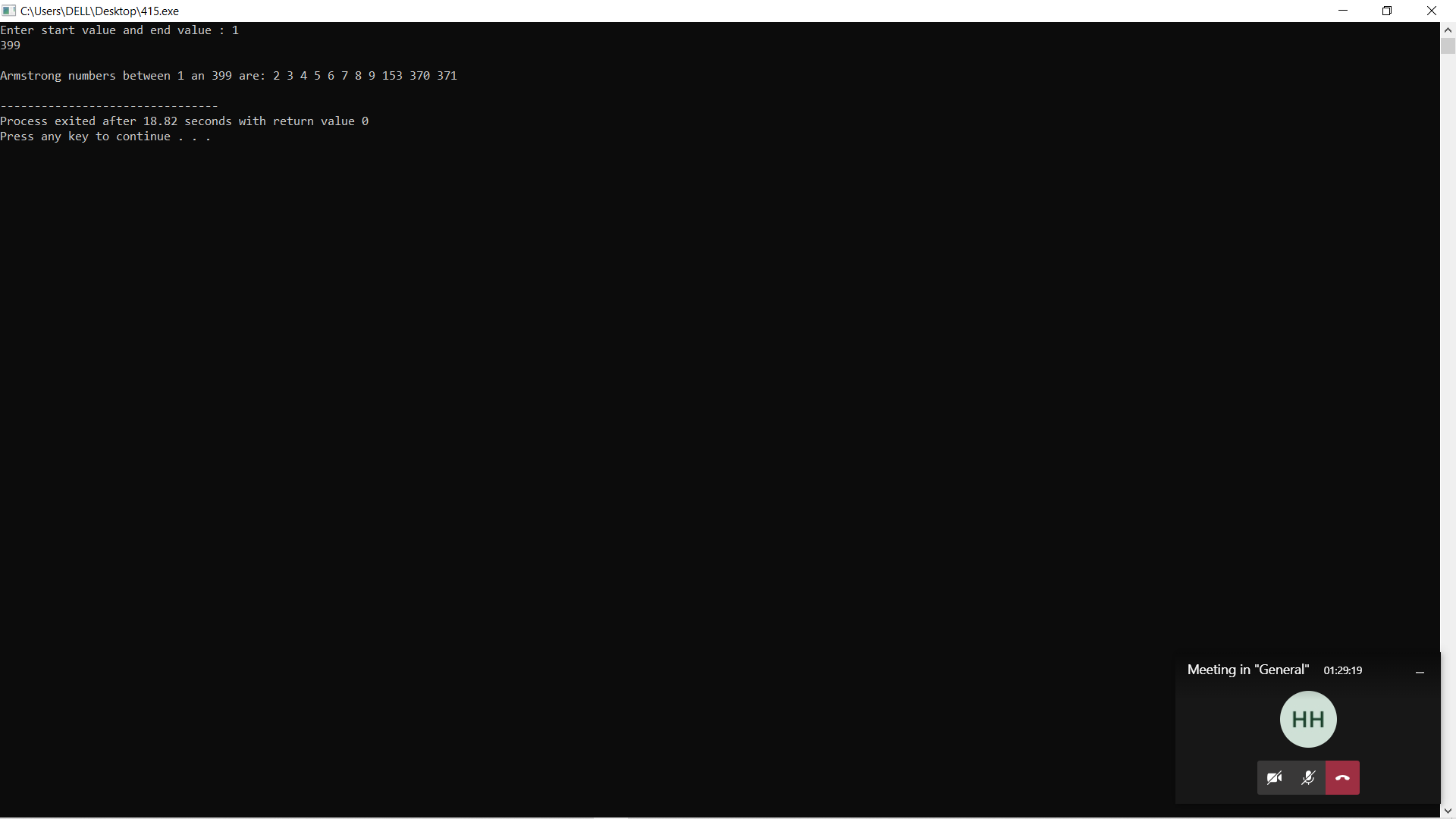
}

printf("\n");

return 0;

}





6.Display factors of a number

#include <stdio.h>

int main() {

int num, i;

printf("Enter a positive integer: ");

scanf("%d", &num);

printf("Factors of %d are: ", num);

for (i = 1; i <= num; ++i) {

if (num % i == 0) {

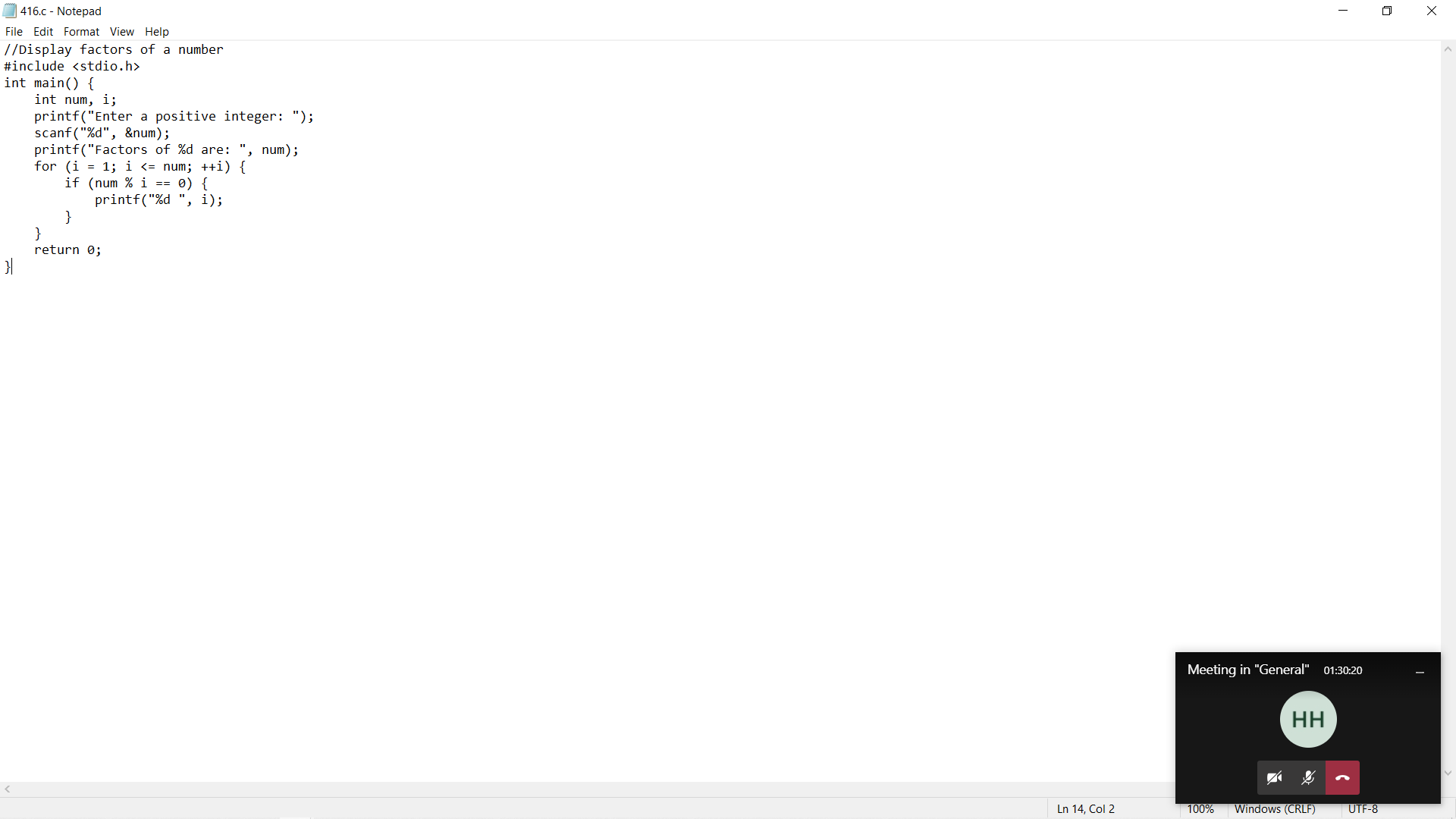
printf("%d ", i);

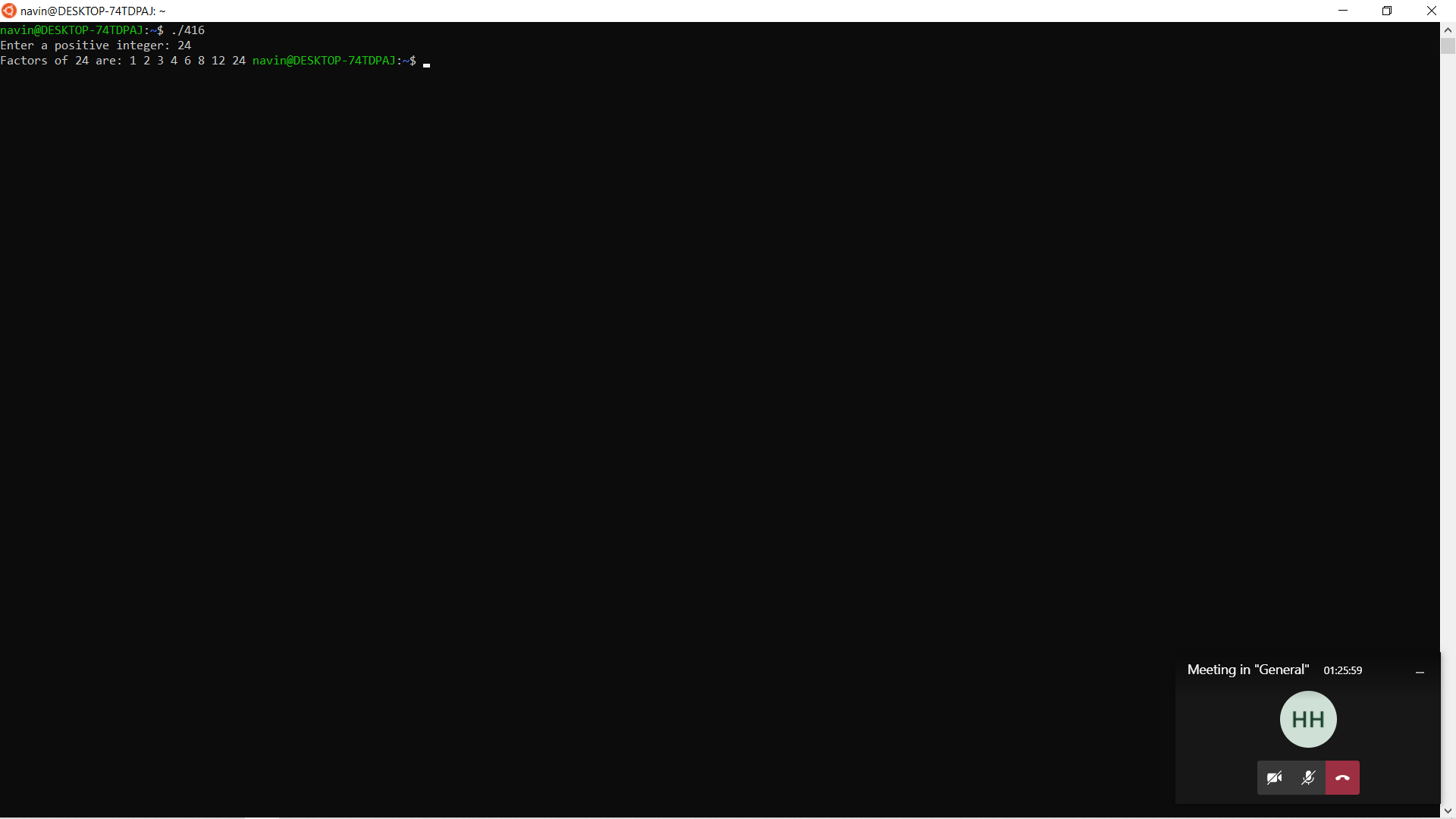
}

}

return 0;

}





7.Print pyramids and triangles

#include <stdio.h>

int main() {

int i, space, rows, k = 0;

printf("Enter the number of rows: ");

scanf("%d", &rows);

for (i = 1; i <= rows; ++i, k = 0) {

for (space = 1; space <= rows - i; ++space) {

printf(" ");

}

while (k != 2 \* i - 1) {

printf("\* ");

++k;

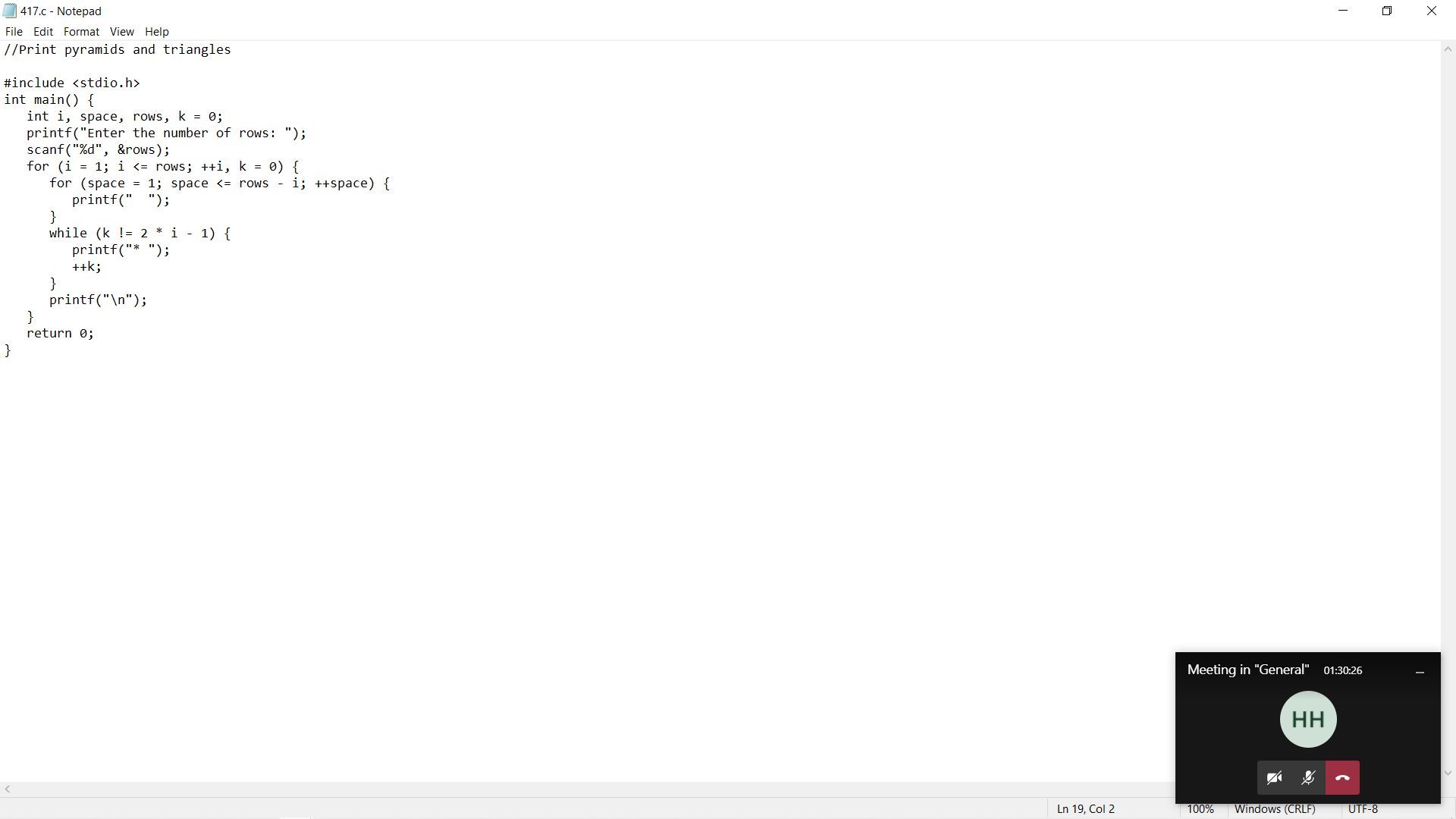
}

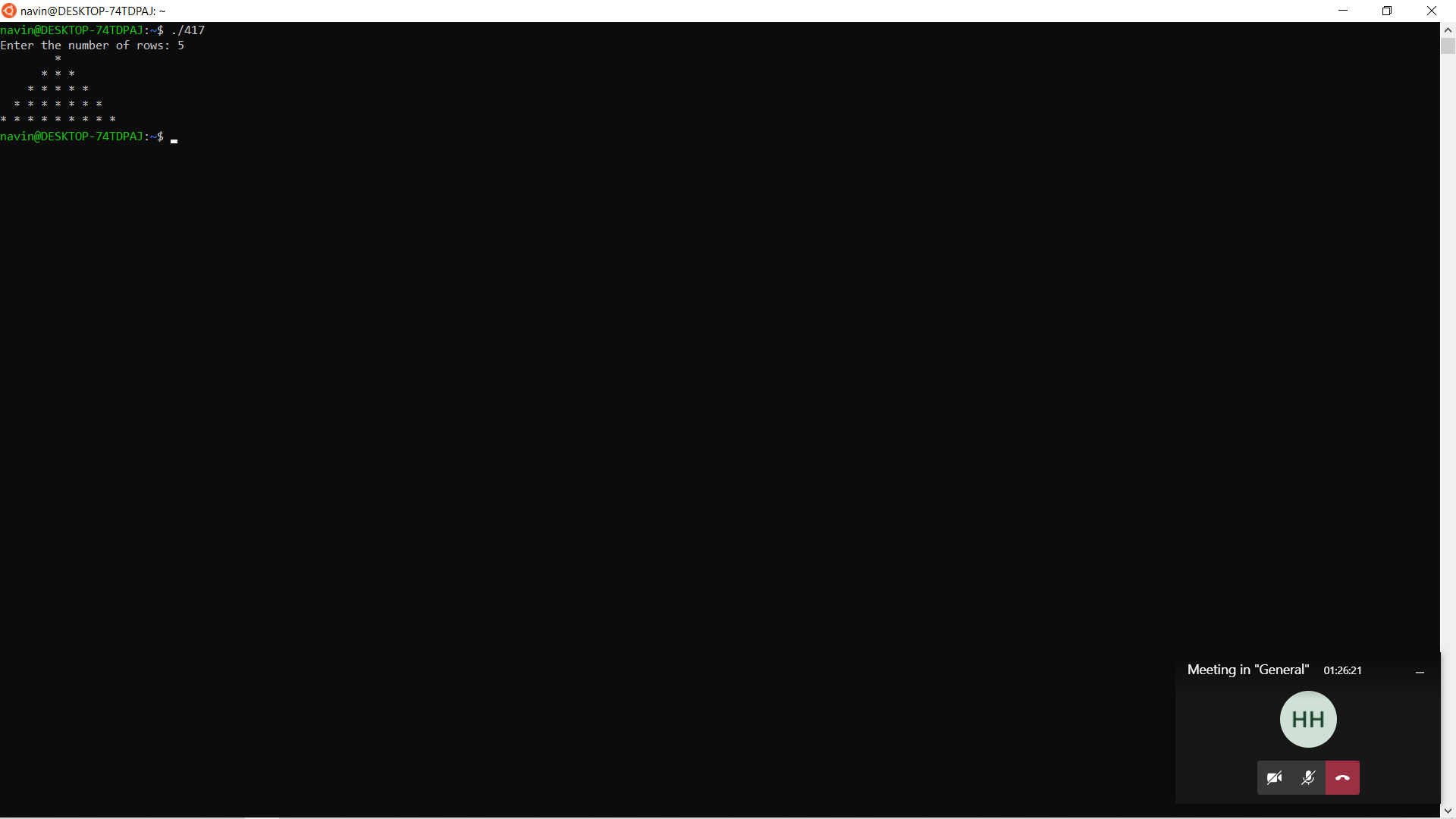
printf("\n");

}

return 0;

}





8.Create a simple calculator

#include <stdio.h>

int main() {

char operator;

double first, second;

printf("Enter an operator (+, -, \*,): ");

scanf("%c", &operator);

printf("Enter two operands: ");

scanf("%lf %lf", &first, &second);

switch (operator) {

case '+':

printf("%.1lf + %.1lf = %.1lf", first, second, first + second);

break;

case '-':

printf("%.1lf - %.1lf = %.1lf", first, second, first - second);

break;

case '\*':

printf("%.1lf \* %.1lf = %.1lf", first, second, first \* second);

break;

case '/':

printf("%.1lf / %.1lf = %.1lf", first, second, first / second);

break;

// operator doesn't match any case constant

default:

printf("Error! operator is not correct");

}

return 0;

}

