**Embedded Systems Lab**

**Assignment 4**

**Name** – Neelangshu Roy

**Reg No** – 20214060

**Group** – C1

Assembly language programs in EdSim51 :

Q1. Assembly code to find average of all numbers stored in an array.

; length of array stored at address 60h

; array elements are present from address 70h onwards

; result is stored at address 50h

mov a, 60h

mov r0, #70h

mov r1, #00h

loop: mov b, a

mov a, r1

add a, @r0

mov r1, a

inc r0

mov a, b

dec a

jnz loop

mov b, 60h

mov a, r1

div ab

mov 50h, a

end

Q2. Assembly program to count no. of 1's in a number(in decimal format) and store the result at an address.

; number is stored at address 30h

; result is stored at address 40h

mov r0, 30h

mov r1, #00h

loop: mov b, #0ah

mov a, r0

div ab

mov r2, a

mov a, b

dec a

jnz go\_on

inc r1

go\_on: mov a, r2

mov r0, a

subb a, #0ah

jnc loop

mov a, r0

dec a

jnz exit

inc r1

exit: mov 40h, r1

end

Q3. Assembly program to check if number is a palindrome(in decimal format) using stack.

; number is at address 30h

; if no .is palindrome value at address 40h is set to 01 from zero

mov r1, #00

mov a, 30h

subb a, #01h

jc jump1

inc r1

ljmp jump

jump1: ljmp setit

jump: inc a

mov 40h, #00

loop: mov b, #10

div ab

subb a, #01h ; carry flag is raised by subb, NOT dec !!!

jc exit

inc r1

inc a

ljmp loop

exit: mov b, #02

mov a, r1

div ab

; mov r0, a

mov r3, 30h

loop2: mov r0, a

mov b, #10

mov a, r3

div ab

push 0f0h

mov r3, a

mov a, r0

dec a

jnz loop2

mov a, r1

anl a, #01h

jz go\_on

mov a, r3

mov b, #10

div ab

mov r3, a

go\_on: mov b, #02

mov a, r1

div ab

; mov r0, a

loop3: mov r0, a

mov b, #10

mov a, r3

div ab

mov r3, a

pop 0e0h

subb a, b

jnz getout

mov a, r0

dec a

jnz loop3

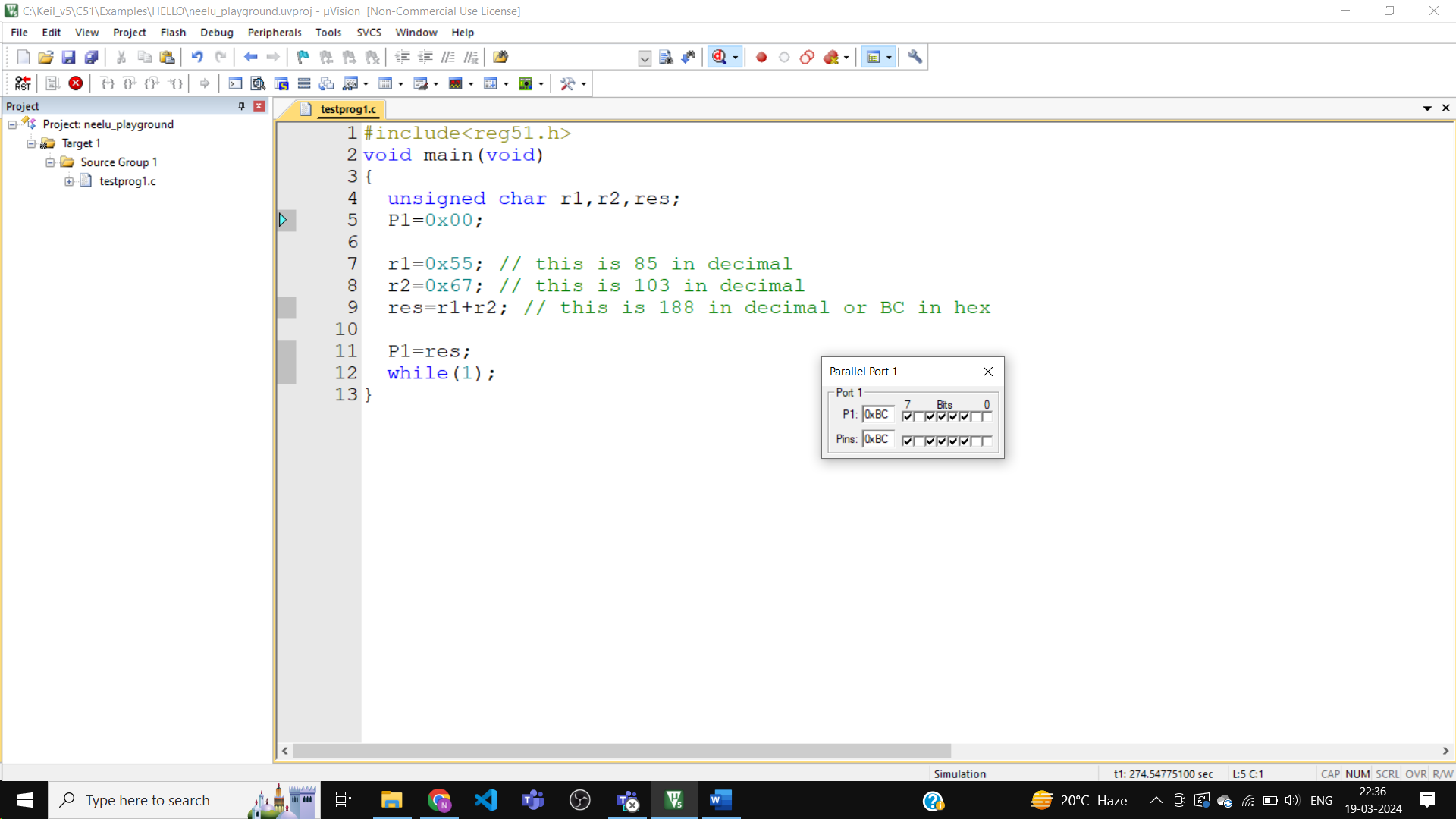
setit: mov 40h, #01

getout: nop

end

Write C programs in Keil :

Q1. Addition of two 8-bit numbers and show output on port P1.



C code :

#include<reg51.h>

void main(void)

{

unsigned char r1,r2,res;

P1=0x00;

r1=0x55; // this is 85 in decimal

r2=0x67; // this is 103 in decimal

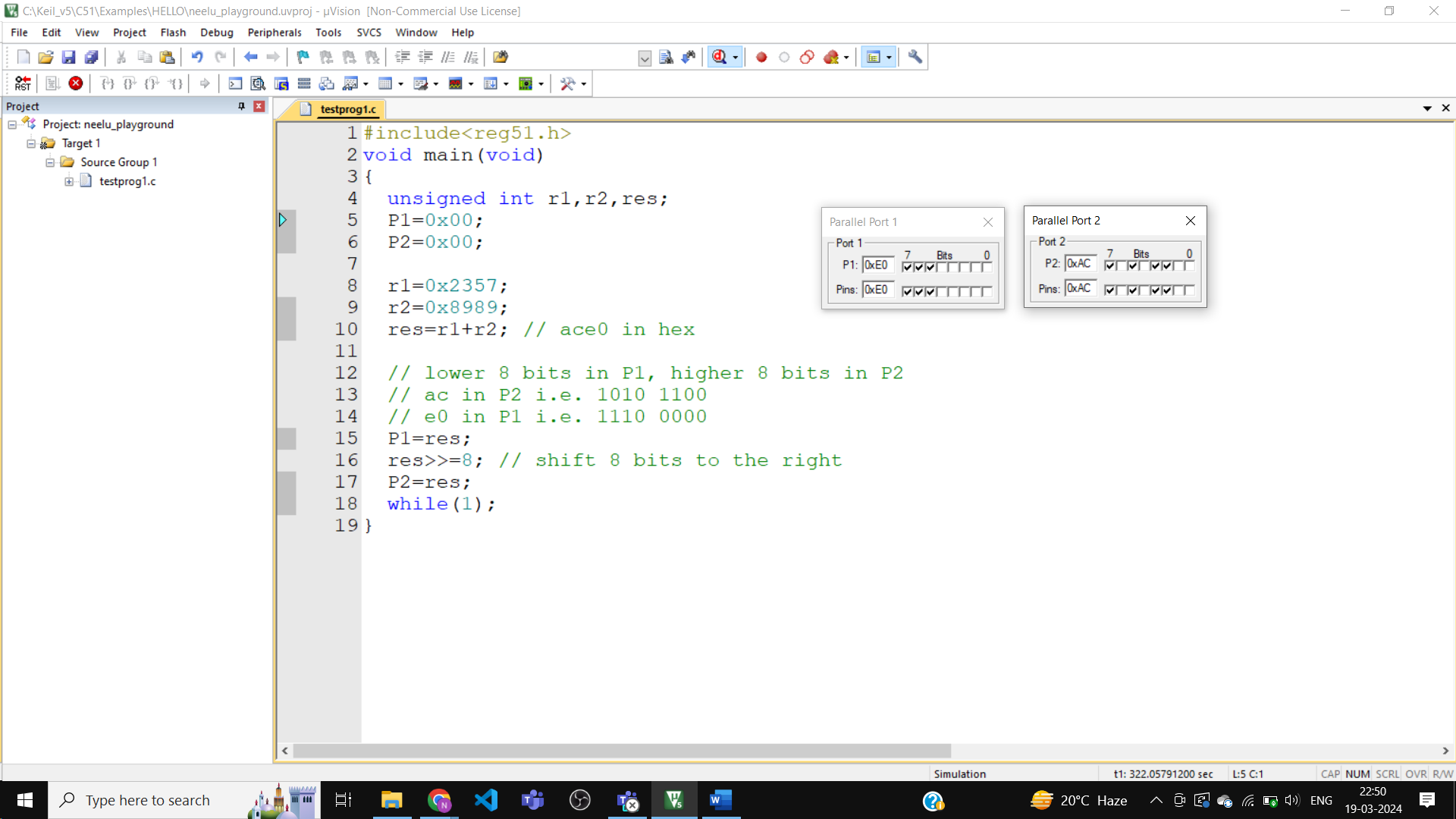
res=r1+r2; // this is 188 in decimal or BC in hex

P1=res;

while(1);

}

Q2. Addition of two 16-bit numbers and show output on ports P1 and P2.



C code :

#include<reg51.h>

void main(void)

{

unsigned int r1,r2,res;

P1=0x00;

P2=0x00;

r1=0x2357;

r2=0x8989;

res=r1+r2; // ace0 in hex

// lower 8 bits in P1, higher 8 bits in P2

// ac in P2 i.e. 1010 1100

// e0 in P1 i.e. 1110 0000

P1=res;

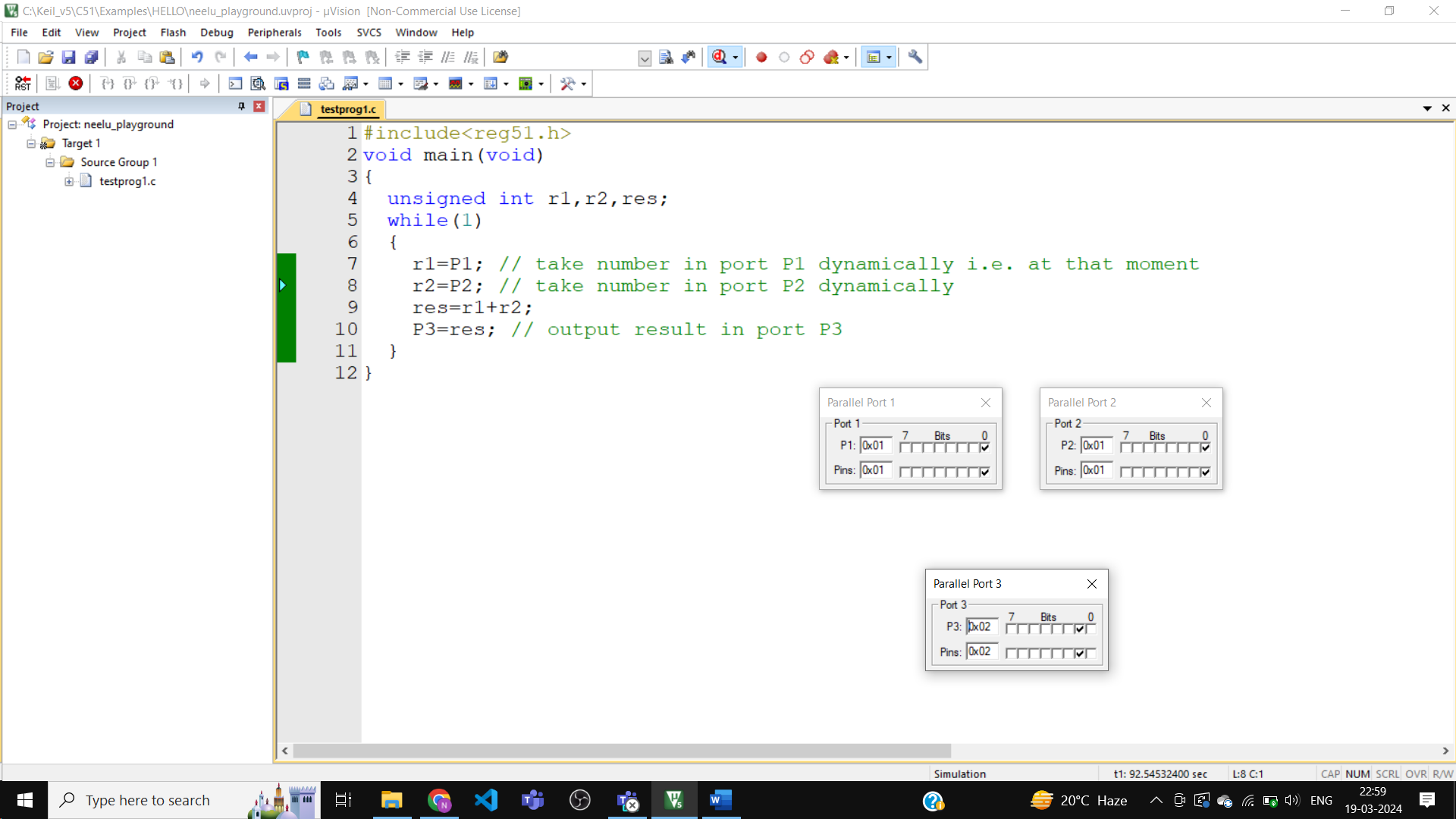
res>>=8; // shift 8 bits to the right

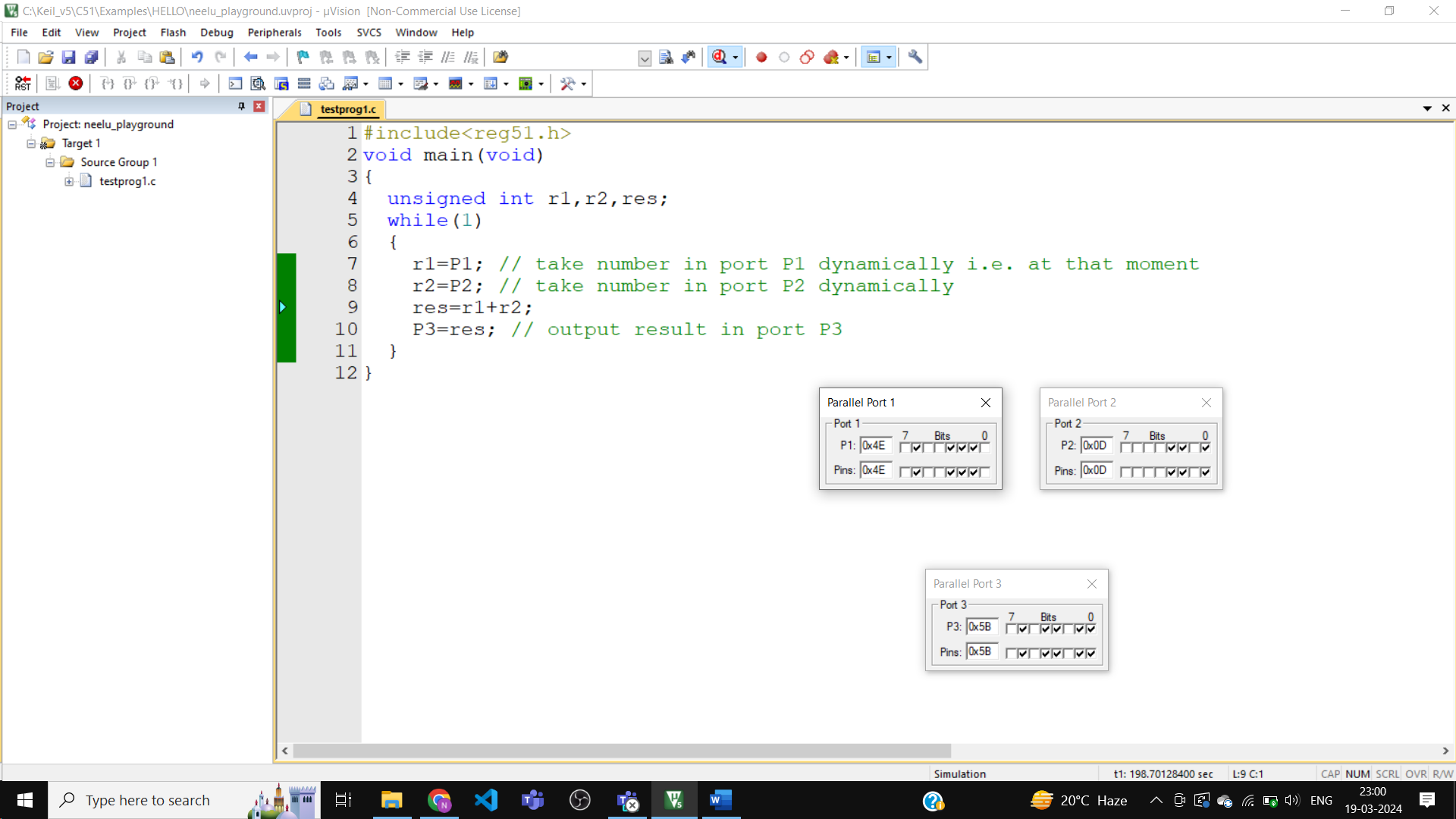
P2=res;

while(1);

}

Q3. Addition of two 8-bit numbers. Take input from ports P1 and P2 and show result on port P3.





C code :

#include<reg51.h>

void main(void)

{

unsigned int r1,r2,res;

while(1)

{

r1=P1; // take number in port P1 dynamically i.e. at that moment

r2=P2; // take number in port P2 dynamically

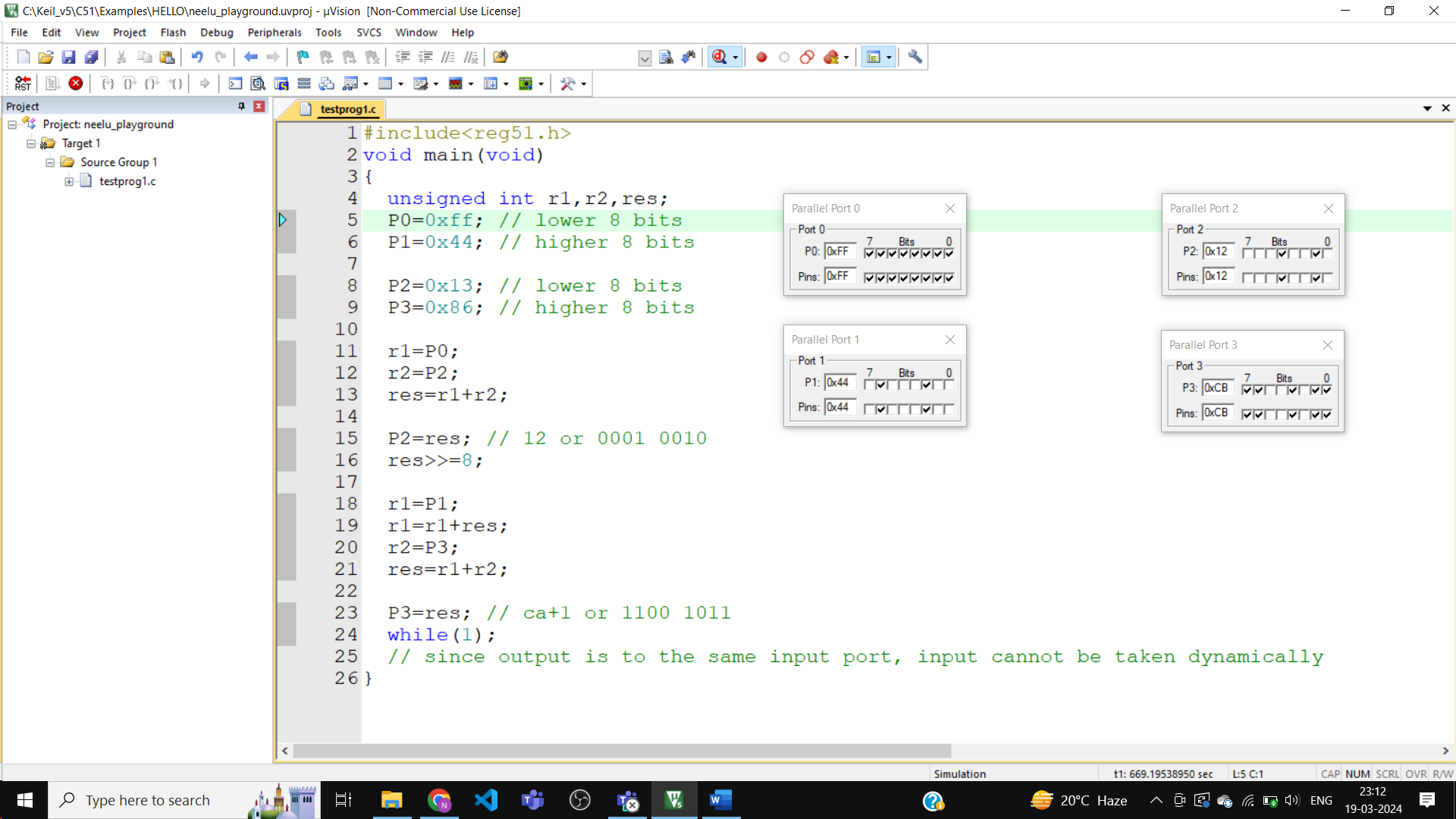
res=r1+r2;

P3=res; // output result in port P3

}

}

Q4. Addition of two 16-bit numbers. Input from ports P0,P1,P2,P3. Output on ports P2 and P3.



C code :

#include<reg51.h>

void main(void)

{

unsigned int r1,r2,res;

P0=0xff; // lower 8 bits

P1=0x44; // higher 8 bits

P2=0x13; // lower 8 bits

P3=0x86; // higher 8 bits

r1=P0;

r2=P2;

res=r1+r2;

P2=res; // 12 or 0001 0010

res>>=8;

r1=P1;

r1=r1+res;

r2=P3;

res=r1+r2;

P3=res; // ca+1 or 1100 1011

while(1);

// since output is to the same input port, input cannot be taken dynamically

}