

**Exercise 1:** Programs on Sequential Statements

**Name:** Rahul Karthik S

**Register Number:** 21BEC1851

**Date:** 18-07-2024

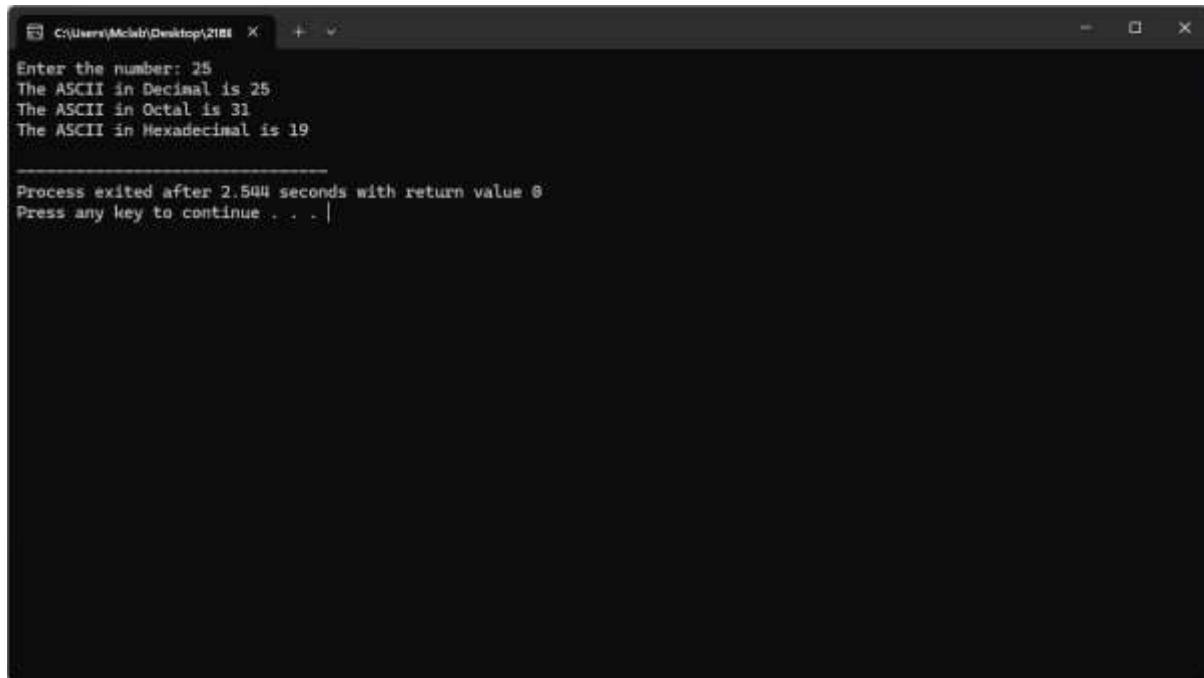
- 1) Write a C Program to display the ASCII code of a character in decimal, octal and hexadecimal.

Code:

```
#include<stdio.h>

int main(){
    int num;
    printf("Enter the number: ");
    scanf("%d", &num);
    printf("The ASCII in Decimal is %d \n", num);
    printf("The ASCII in Octal is %o \n", num);
    printf("The ASCII in Hexadecimal is %x \n", num);
    return 0;
}
```

Output:



```
C:\Users\MeLab\Desktop\21BEC1851 X + v
Enter the number: 25
The ASCII in Decimal is 25
The ASCII in Octal is 31
The ASCII in Hexadecimal is 19
_____
Process exited after 2.504 seconds with return value 0
Press any key to continue . . . |
```

- 2) Write a C program to reverse a 3-digit number.

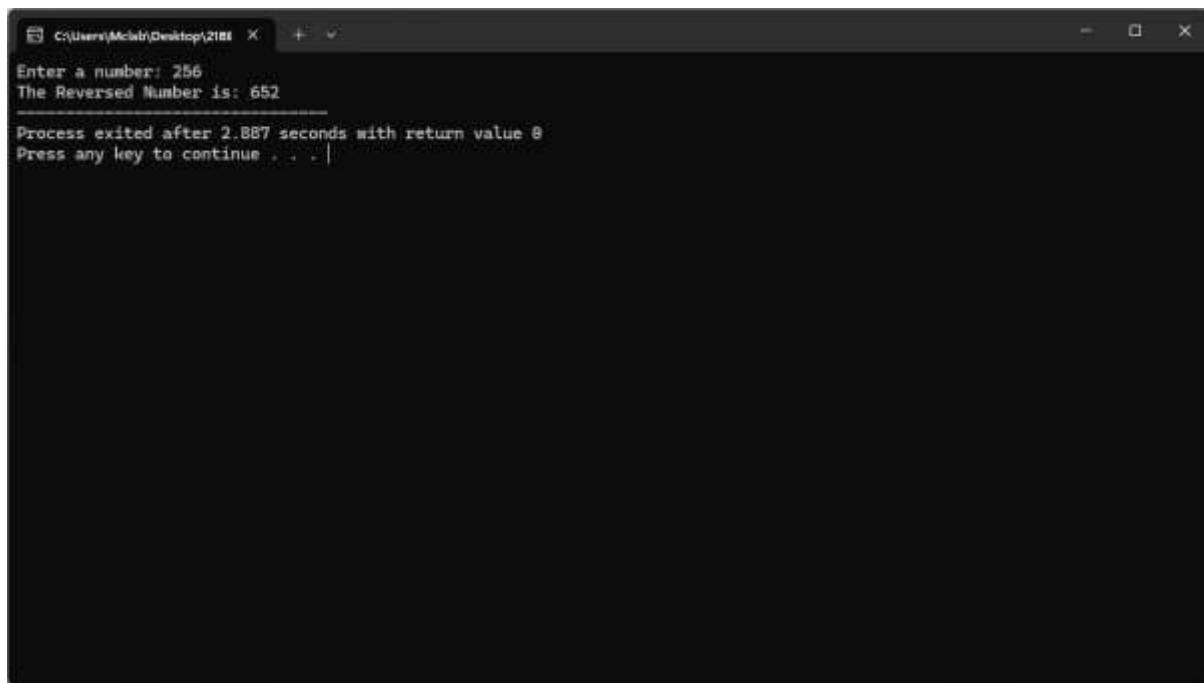
Code:

```
#include<stdio.h>

int main(){
```

```
int num, rev, a, b, c;
printf("Enter a number: ");
scanf("%d", &num);
a = num % 10;
num = num / 10;
b = num % 10;
num = num / 10;
c = num % 10;
rev = a*100 + b*10 + c;
printf("The Reversed Number is: %d", rev);
return 0;
}
```

Output:



```
C:\Users\Melebi\Desktop\2188 X + v
Enter a number: 256
The Reversed Number is: 652
Process exited after 2.887 seconds with return value 0
Press any key to continue . . .
```

- 3) Write a C Program to calculate compound interest.

Code:

```
#include<stdio.h>
#include<math.h>
int main(){
    float p, n, CI, r;
    printf("Enter the Principal: ");
    scanf("%f", &p);
    printf("Enter the Rate of Interest: ");
```

```

        scanf("%f", &r);
        printf("Enter the Number of Years: ");
        scanf("%f", &n);
        CI = p*pow((1+(r/100)), n);
        printf("Compound Interest: %.2f", CI - p);
        return 0;
    }

```

Output:

```

C:\Users\Melebi\Desktop\2188 X + v
Enter the Principal: 15000
Enter the Rate of Interest: 2.5
Enter the Number of Years: 3
Compound Interest: 1153.36
Process exited after 5.020 seconds with return value 0
Press any key to continue . . .

```

- 4) Write a C Program to convert the temperature from Fahrenheit to Celsius.

Code:

```

#include<stdio.h>
#include<math.h>

int main(){
    float f, c;
    printf("Enter the Temperature in Fahrenheit: ");
    scanf("%f", &f);
    c = (f - 32) * 5/9;
    printf("The Temperature in Celcius is %f", c);
    return 0;
}

```

Output:

The screenshot shows a terminal window with the following text:

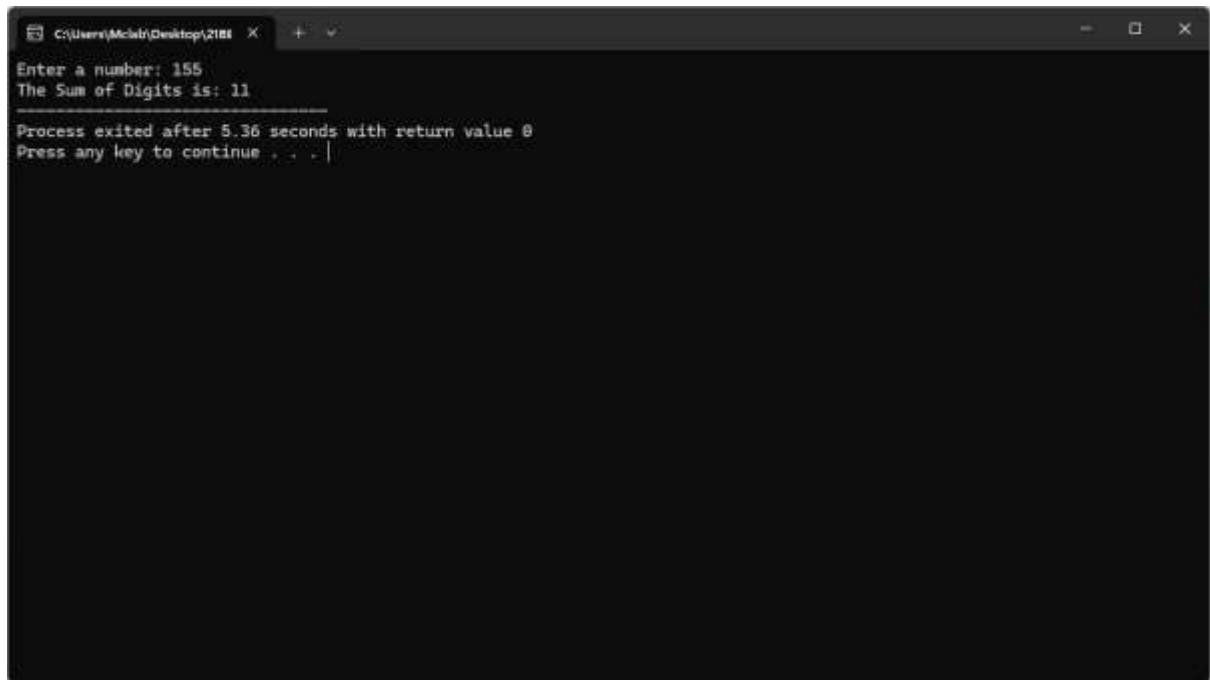
```
C:\Users\Meclab\Desktop\2188 X + v
Enter the Temperature in Fahrenheit: 79
The Temperature in Celcius is 26.111111
Process exited after 52.33 seconds with return value 0
Press any key to continue . . .
```

- 5) Write a C Program to Print the sum of digits in a 3-digit number.

Code:

```
#include<stdio.h>
int main(){
    int num, rev, a, b, c;
    printf("Enter a number: ");
    scanf("%d", &num);
    a = num % 10;
    num = num / 10;
    b = num % 10;
    num = num / 10;
    c = num % 10;
    rev = a + b + c;
    printf("The Sum of Digits is: %d", rev);
    return 0;
}
```

Output:



```
C:\Users\Meclab\Desktop\2188 > Enter a number: 155
The Sum of Digits is: 11
Process exited after 5.36 seconds with return value 0
Press any key to continue . . .
```

6) Write a C Program to find the average of three numbers.

Code:

```
#include<stdio.h>

int main(){

    float a, b ,c;
    float avg;
    printf("Enter the First Number: ");
    scanf("%f", &a);
    printf("Enter the Second Number: ");
    scanf("%f", &b);
    printf("Enter the Third Number: ");
    scanf("%f", &c);
    avg = (a+b+c)/3;
    printf("Average: %.2f", avg);
    return 0;
}
```

Output:

```
C:\Users\MeLab\Desktop\2188 + v
Enter the First Number: 18
Enter the Second Number: 15
Enter the Third Number: 20
Average: 15.66
Process exited after 3.084 seconds with return value 0
Press any key to continue . . . |
```

7) Write a C Program to calculate perimeter of the rectangle.

Code:

```
#include<stdio.h>
int main(){
    float l, b;
    float peri;
    printf("Enter the Length: ");
    scanf("%f", &l);
    printf("Enter the Breadth: ");
    scanf("%f", &b);
    peri = (l+b)*2;
    printf("Perimeter of the Rectangle: %.2f", peri);
    return 0;
}
```

Output:

```
C:\Users\Meclab\Desktop\2188 X + v
Enter the Length: 12
Enter the Breadth: 9
Perimeter of the Rectangle: 42.00
Process exited after 7.234 seconds with return value 0
Press any key to continue . . . |
```

8) Write a C program to calculate volume of a cone.

Code:

```
#include<stdio.h>
#include<math.h>
int main(){
    float v, h, r;
    const float pi = 3.14;
    printf("Enter the Height: ");
    scanf("%f", &h);
    printf("Enter the Radius: ");
    scanf("%f", &r);
    v = (0.33)*h*pi*pow(r, 2);
    printf("The volume of the cone is %f", v);
    return 0;
}
```

Output:

The screenshot shows a terminal window with the following text:

```
C:\Users\Rahul Karthik\Desktop\21BEC1851\Ex1.exe
Enter the Height: 25
Enter the Radius: 10
The volume of the cone is 2590.500000
Process exited after 10.96 seconds with return value 0
Press any key to continue . . .
```

9) Write a C program to calculate total surface area of a cylinder.

Code:

```
#include<stdio.h>
#include<math.h>
int main(){
    float a, h, r;
    const float pi = 3.14;
    printf("Enter the Height: ");
    scanf("%f", &h);
    printf("Enter the Radius: ");
    scanf("%f", &r);
    a = 2*pi*r*h + 2*pi*pow(r,2);
    printf("The surface area of the cylinder is %f", a);
    return 0;
}
```

Output:

```
C:\Users\Rahul Karthik\Desktop\21BEC1851\Ex1.exe
Enter the Height: 25
Enter the Radius: 10
The surface area of the cylinder is 2198.000000
Process exited after 3.16 seconds with return value 0
Press any key to continue . . .
```

10) Write a C program to find the absolute value of a number using ternary operator.

Code:

```
#include<stdio.h>
#include<math.h>
int main(){
    int num, abs;
    printf("Enter the Number: ");
    scanf("%d", &num);
    abs = (num>=0)?num:-num;
    printf("The absolute value of %d is %d", num, abs);
    return 0;
}
```

Output:

```
C:\Users\Rahul Karthik\Desktop\21BEC1851\Ex1.exe
Enter the Number: -25
The absolute value of -25 is 25
Process exited after 4.714 seconds with return value 0
Press any key to continue . . .
```

11) Write a C program to find the sum of an arithmetic series.

Code:

```
#include<stdio.h>
#include<math.h>
int main(){
    int n;
    float a, d, s;
    printf("Enter the First Term, Common Difference and Number of Terms: /n");
    scanf("%f",&a);
    scanf("%f",&d);
    scanf("%d",&n);
    s = (n/2.0)*(2*a + (n-1)*d);
    printf("The Sum of Arithmetic Progression is: %.2f", s);
    return 0;
}
```

Output:

```
C:\Users\Rahul Karthik\Desktop\21BEC1851\Ex1.exe
Enter the First Term, Common Difference and Number of Terms: 10
10
10
The Sum of Arithmetic Progression is: 550.00
Process exited after 3.631 seconds with return value 0
Press any key to continue . . .
```

12) Write a C program to find the sum of a geometric series.

Code:

```
#include<stdio.h>
#include<math.h>
int main(){
    float n, a, r, s;
    printf("Enter the First Term, Common Ratio and Number of Terms: \n");
    scanf("%f",&a);
    scanf("%f",&r);
    scanf("%f",&n);
    s = (r==1.0)?(a*n):(a*(1-pow(r,n))/(1-r));
    printf("The Sum of Arithmetic Progression is: %.2f", s);
    return 0;
}
```

Output:

C:\Users\Rahul Karthik\Desktop\21BEC1851\Ex1.exe

```
Enter the First Term, Common Ratio and Number of Terms:  
1  
3  
5  
The Sum of Arithmetic Progression is: 121.00  
Process exited after 42.78 seconds with return value 0  
Press any key to continue . . .
```

## **Exercise 2: Programs on Conditional and Control Statements**

**Register Number:** 21BEC1851

**Date:** 25-07-2024

- 1) C Program to check whether a given character is upper case, lower case, or special character using else-if ladder.

Code:

```
#include<stdio.h>

int main(){

    char ch;

    printf("Enter the character: ");

    scanf("%c", &ch);

    if(ch >= 'A' && ch <= 'Z')

        printf("The character is a Uppercase.");

    else if (ch >= 'a' && ch <= 'z')

        printf("The character is a Lowercase.");

    else if(ch >= '1' && ch <= '9')

        printf("The character is a Number.");

    else

        printf("The character is a special character.");

    return 0;
}
```

Output:

```
C:\Users\Mclab\Desktop\21BE X + - □ ×  
Enter the character: A  
The character is a Uppercase.  
-----  
Process exited after 2.623 seconds with return value 0  
Press any key to continue . . . |
```

```
C:\Users\Mclab\Desktop\21BE X + - □ ×  
Enter the character: a  
The character is a Lowercase.  
-----  
Process exited after 2.264 seconds with return value 0  
Press any key to continue . . . |
```

```
C:\Users\Mclab\Desktop\2IBE X + | - □ ×  
Enter the character: 1  
The character is a Number.  
-----  
Process exited after 1.575 seconds with return value 0  
Press any key to continue . . . |
```

```
C:\Users\Mclab\Desktop\2IBE X + | - □ ×  
Enter the character: /  
The character is a special character.  
-----  
Process exited after 1.964 seconds with return value 0  
Press any key to continue . . . |
```

2) C Program to check if a given year is a leap year or not.

Code:

```
#include<stdio.h>  
  
int main(){  
  
    int year;  
  
    printf("Enter the year: ");
```

```
scanf("%d", &year);

if (year%4 == 0 && (year%100 != 0 || year%400 == 0))

    printf("It is a leap year!");

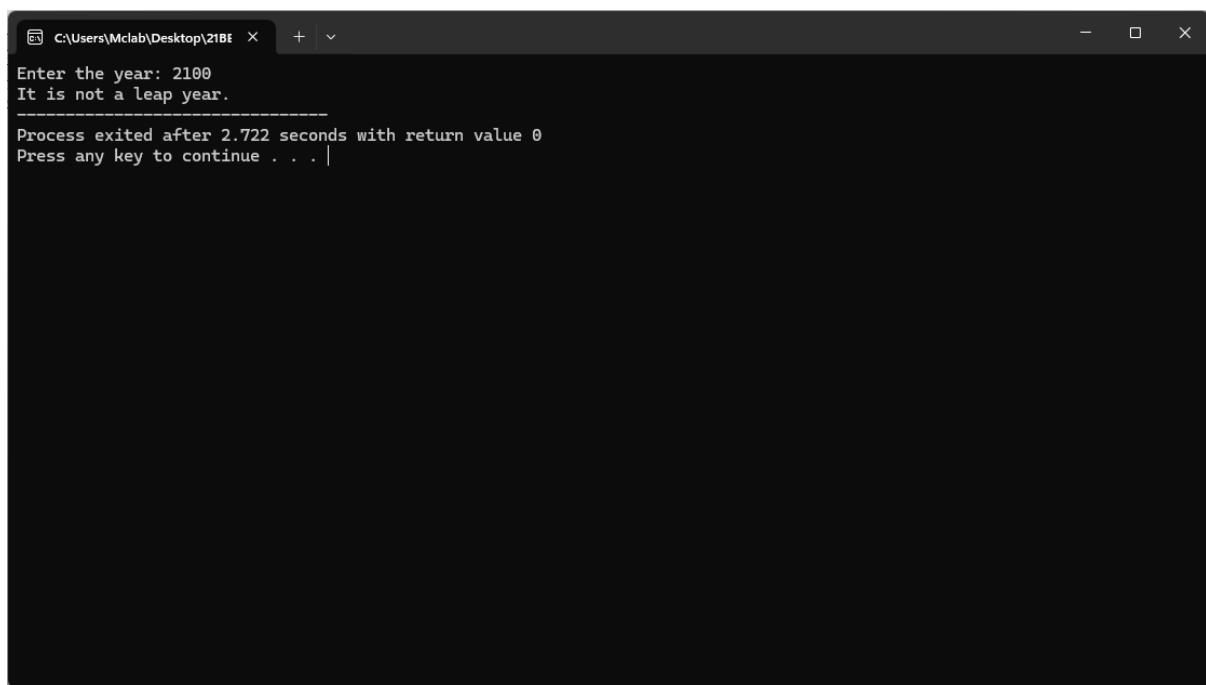
else

    printf("It is not a leap year.");

return 0;

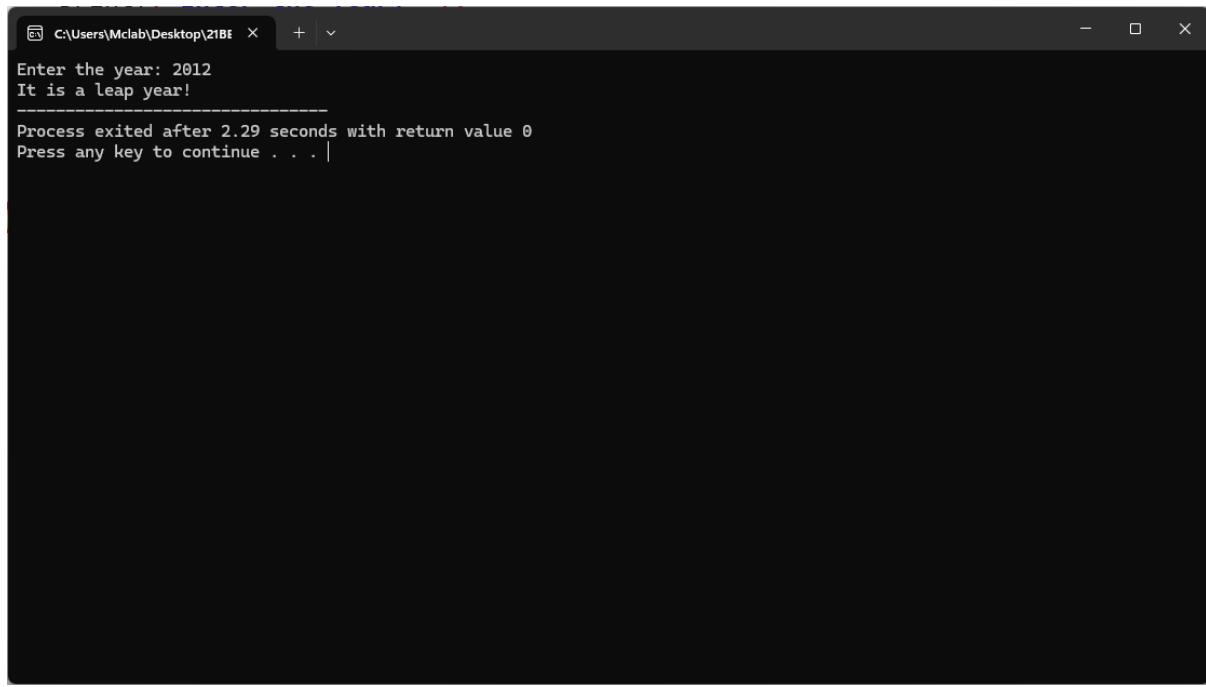
}
```

Output:



```
C:\Users\MyLab\Desktop\21B X + - □ ×

Enter the year: 2100
It is not a leap year.
-----
Process exited after 2.722 seconds with return value 0
Press any key to continue . . . |
```



```
C:\Users\Mclab\Desktop\21BE X + v
Enter the year: 2012
It is a leap year!
-----
Process exited after 2.29 seconds with return value 0
Press any key to continue . . . |
```

- 3) C Program to print Fibonacci series upto nth term using do-while loop. Take only positive number as input.

Code:

```
#include<stdio.h>

int main() {

    int n, t1 = 0, t2 = 1, nxt, i;

    printf("Enter a positive number: ");

    scanf("%d", &n);

    while (n <= 0) {

        printf("Please enter a positive number: ");

        scanf("%d", &n);
    }

    printf("Fibonacci Series: ");

    i = 1;

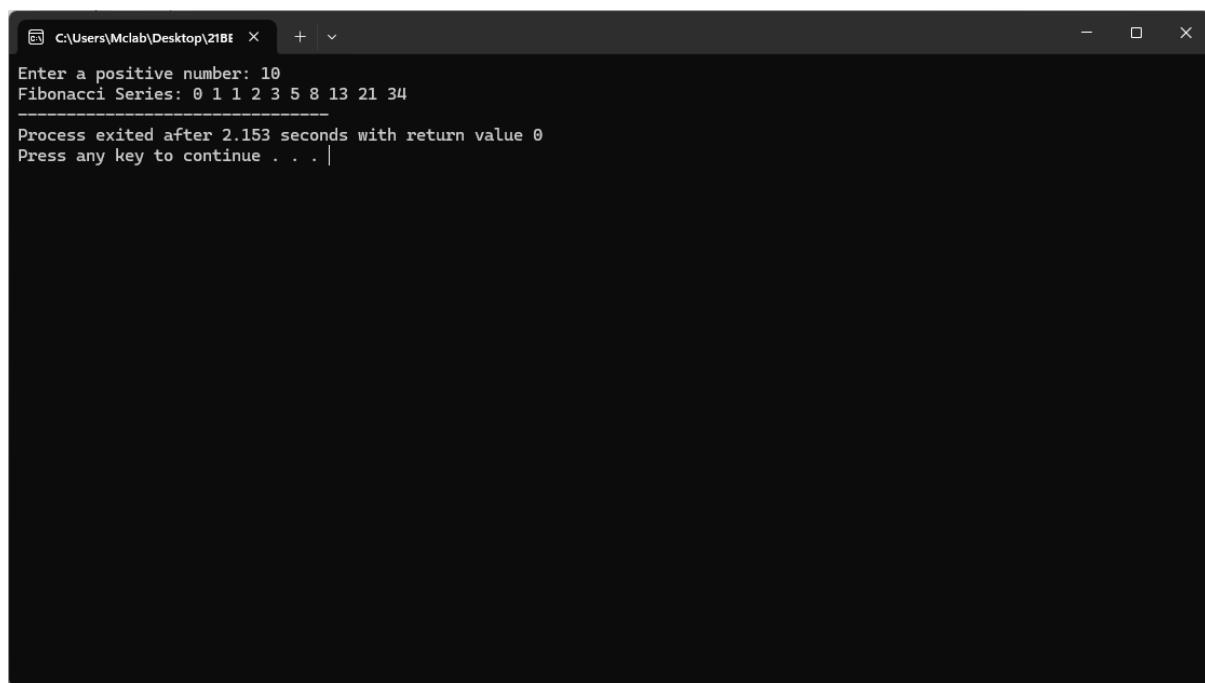
    do {

        if (i == 1) {

            printf("%d ", t1);
```

```
    } else if (i == 2) {  
  
        printf("%d ", t2);  
  
    }  
  
else {  
  
    nxt = t1 + t2;  
  
    t1 = t2;  
  
    t2 = nxt;  
  
    printf("%d ", nxt);  
  
}  
  
    i++;  
  
}  
  
while (i <= n);  
  
return 0;  
}
```

Output:



The screenshot shows a terminal window titled 'C:\Users\Mylab\Desktop\21BF'. The window displays the following text:  
Enter a positive number: 10  
Fibonacci Series: 0 1 1 2 3 5 8 13 21 34  
-----  
Process exited after 2.153 seconds with return value 0  
Press any key to continue . . . |

- 4) C Program to print the day name for a given day number using switch-case.

Code:

```
#include<stdio.h>

int main(){
    int n;
    printf("Enter the Day Number (1-7): ");
    scanf("%d", &n);
    switch(n){

        case 1:
            printf("Monday");
            break;

        case 2:
            printf("Tuesday");
            break;

        case 3:
            printf("Wednesday");
            break;

        case 4:
            printf("Thursday");
            break;

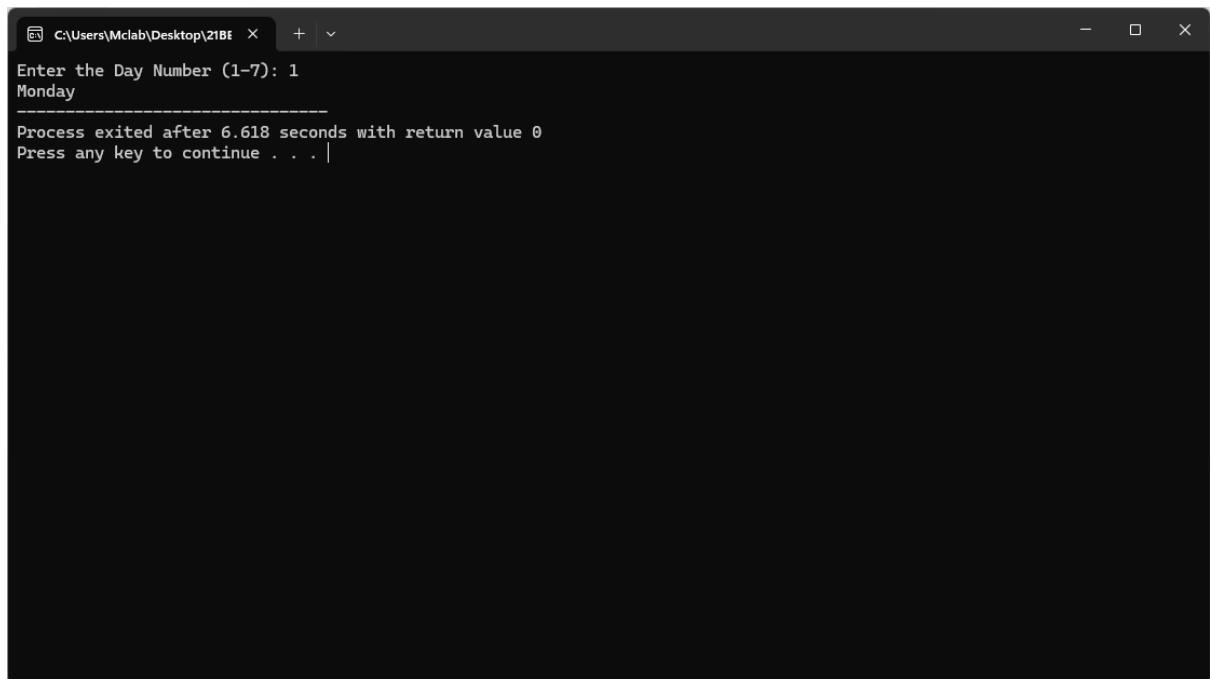
        case 5:
            printf("Friday");
            break;

        case 6:
            printf("Saturday");
            break;

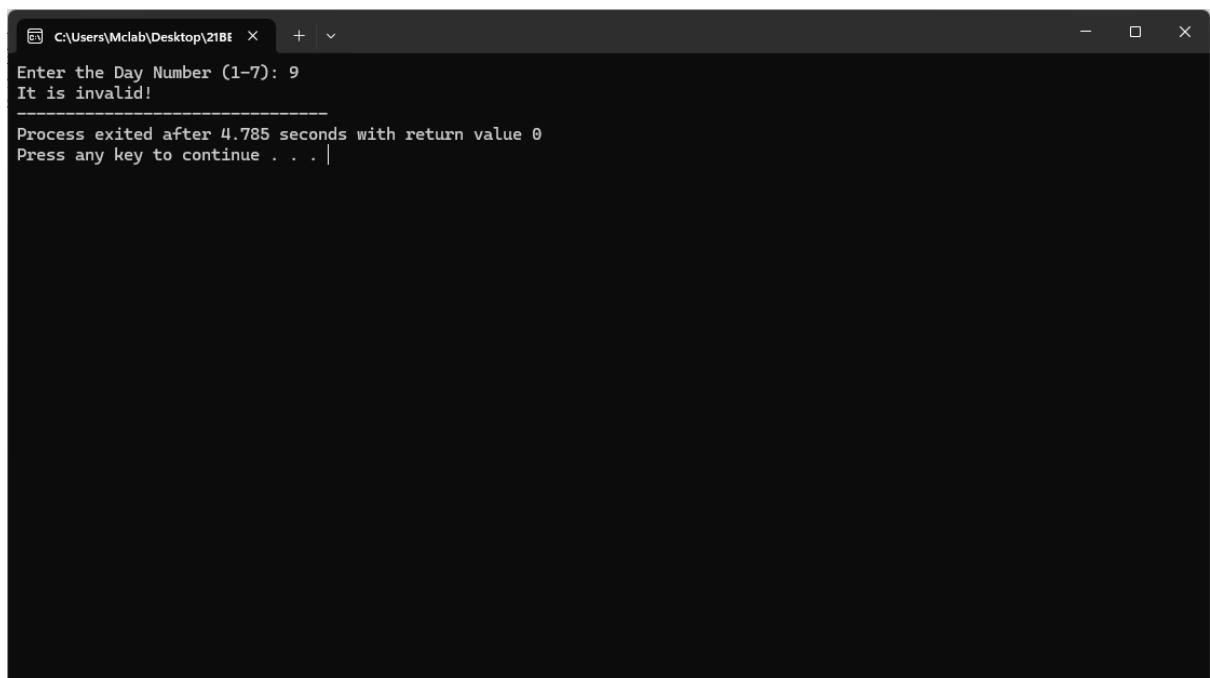
        case 7:
            printf("Sunday");
    }
}
```

```
        break;  
  
    default:  
  
        printf("It is invalid!");  
  
    }  
  
}
```

Output:



```
C:\Users\Mclab\Desktop\21BF > Enter the Day Number (1-7): 1  
Monday  
-----  
Process exited after 6.618 seconds with return value 0  
Press any key to continue . . . |
```



```
C:\Users\Mclab\Desktop\21BF > Enter the Day Number (1-7): 9  
It is invalid!  
-----  
Process exited after 4.785 seconds with return value 0  
Press any key to continue . . . |
```

5) C Program to print the factorial of a given number.

Code:

```
#include<stdio.h>

int main(){

    int n, i, fac = 1;

    printf("Enter the Number: ");

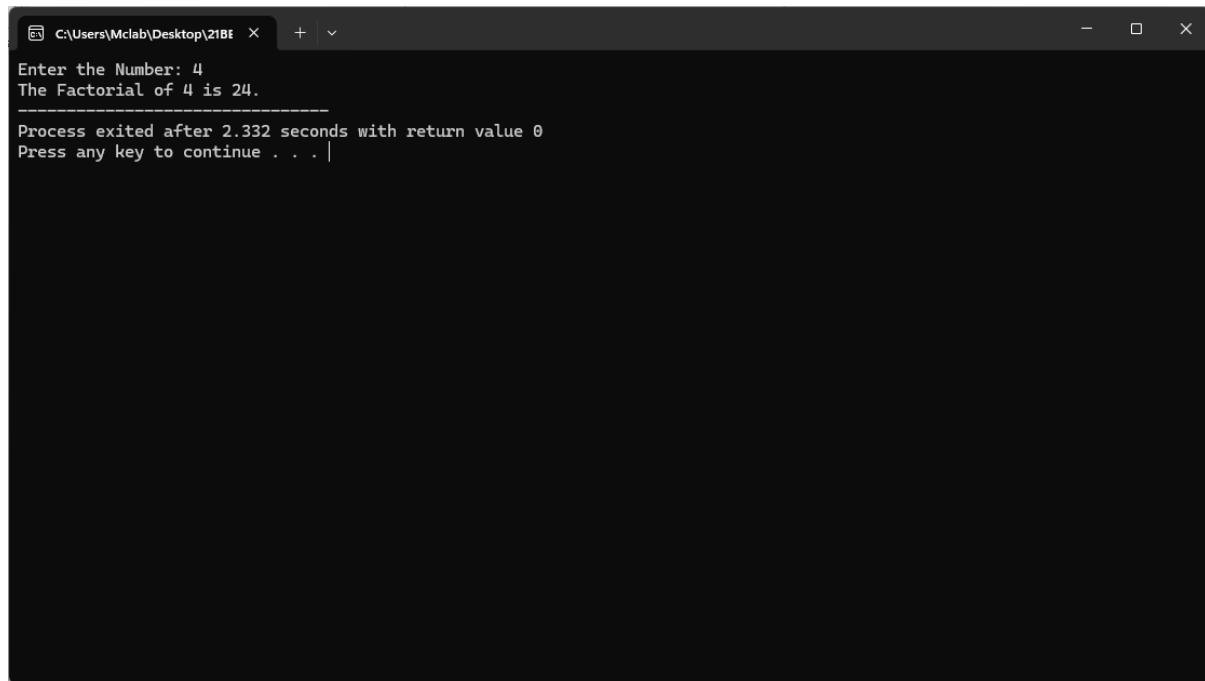
    scanf("%d", &n);

    for(i = 1; i <= n; i++){
        fac *= i;
    }

    printf("The Factorial of %d is %d.", n, fac);

    return 0;
}
```

Output:



The screenshot shows a terminal window titled 'C:\Users\Mclab\Desktop\21B' with the following output:

```
Enter the Number: 4
The Factorial of 4 is 24.
-----
Process exited after 2.332 seconds with return value 0
Press any key to continue . . . |
```

6) C Program to print number from 1 to 100 not divisible by 2 or 7. Use continue statement.

Code:

```
#include<stdio.h>
```

```
int main(){

    int i;

    for (i=1; i<=100; i++){

        if(i%2 == 0 || i%7 ==0)

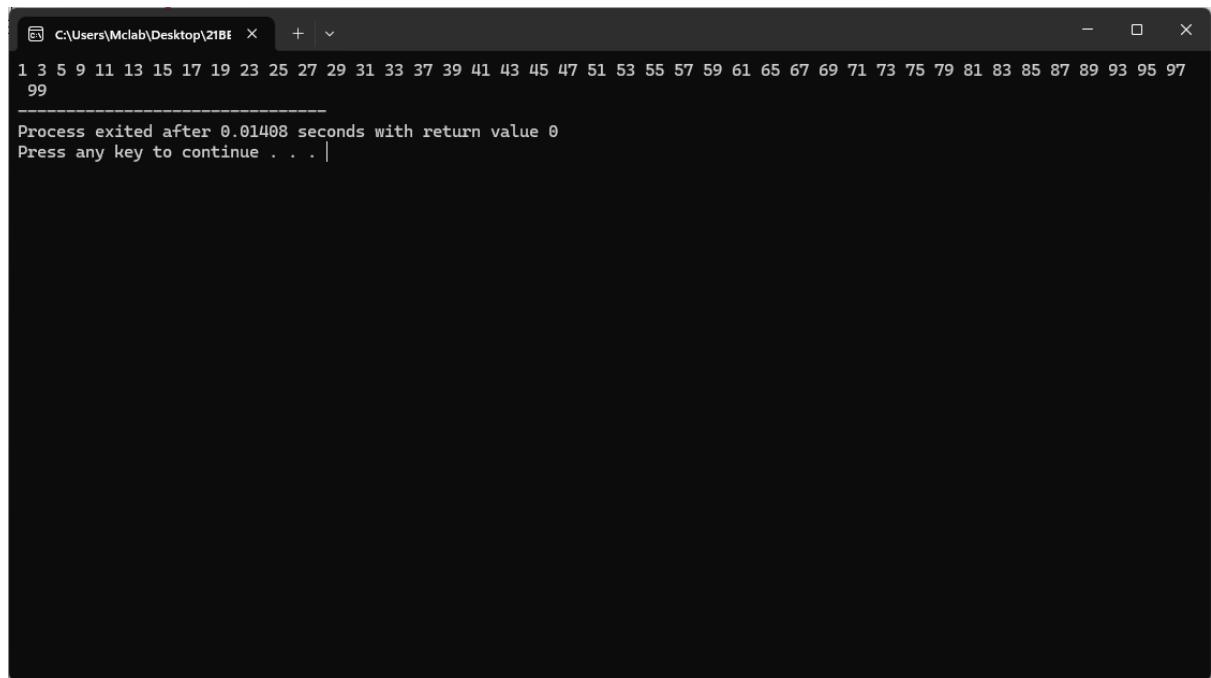
            continue;

        printf("%d ", i);

    }

}
```

Output:



```
C:\Users\McLab\Desktop\2IBE + v
1 3 5 9 11 13 15 17 19 23 25 27 29 31 33 37 39 41 43 45 47 51 53 55 57 59 61 65 67 69 71 73 75 79 81 83 85 87 89 93 95 97
99
-----
Process exited after 0.01408 seconds with return value 0
Press any key to continue . . . |
```

Challenges:

- 1) While purchasing certain items, a discount of 10% is offered if the quantity purchased is more than 1000. If quantity and price per item are input through keyboard. Write a C Program to calculate the total expenses.

Code:

```
#include<stdio.h>

int main(){

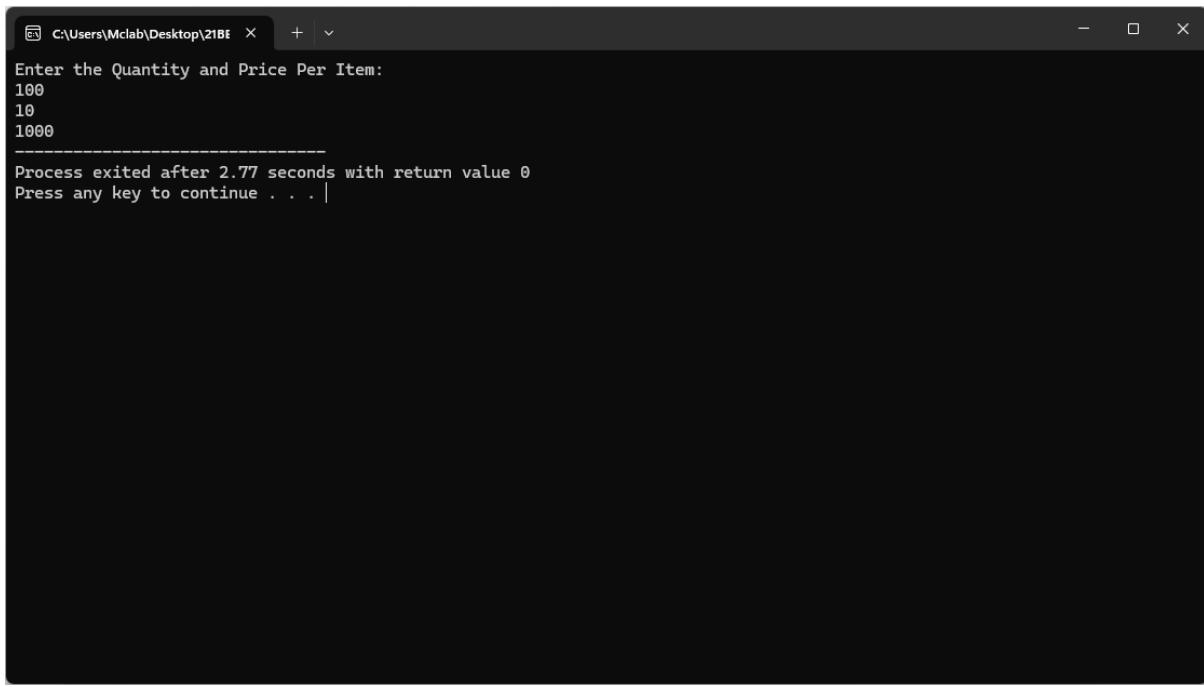
    int q, price, tp;

    printf("Enter the Quantity and Price Per Item: \n");
```

```
scanf("%d", &q);
scanf("%d", &price);
if (q >= 1000){
    tp = q*price - 0.1*(q*price);
}
else
    tp = q*price;
printf("%d", tp);
return 0;
}
```

Output:

```
C:\Users\Mclab\Desktop\21BE
Enter the Quantity and Price Per Item:
1000
10
9000
-----
Process exited after 3.316 seconds with return value 0
Press any key to continue . . . |
```



```
C:\Users\Mclab\Desktop\21BE + - ×
Enter the Quantity and Price Per Item:
100
10
1000
Process exited after 2.77 seconds with return value 0
Press any key to continue . . . |
```

- 2) An electric power distribution company charges its domestic consumers as follows:

#### Consumption Units Rate of Charge

- 0-200 Rs. 0.50 per unit
- 201-400 Rs. 100 plus Rs. 0.65 per unit excess of 200
- 401-600 Rs. 230 plus Rs. 0.80 per unit excess of 400
- 601 and above Rs. 390 plus Rs. 1.00 per unit excess of 600

Write the C program to take the number of units and the customer name as input and print the charge.

Code:

```
#include<stdio.h>

int main(){

    int units, total = 0;

    printf("Enter the total Units Consumed: ");

    scanf("%d", &units);

    if (units >= 0 && units <= 200)

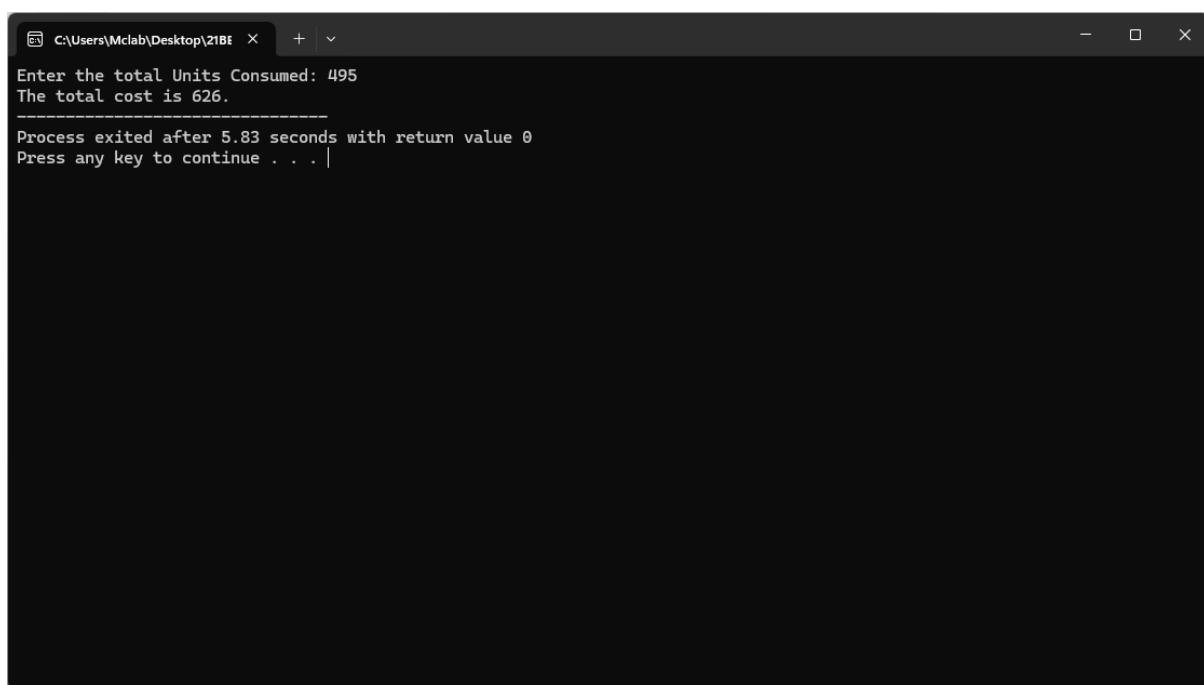
        total = 0.5*units;

    else if (units >= 201 && units <= 400)

        total = 100 + 0.65*units;
```

```
else if (units >= 401 && units <= 600)
    total = 230 + 0.80*units;
else
    total = 390 + units;
printf("The total cost is %d.", total);
return 0;
}
```

Output:



```
C:\Users\... X + - □ ×
Enter the total Units Consumed: 495
The total cost is 626.
-----
Process exited after 5.83 seconds with return value 0
Press any key to continue . . . |
```

### **Exercise 3: Programs on Arrays**

**Register Number:** 21BEC1851

**Date:** 01-08-2024

- 1) Write a C program to search an element in an array. Initialize array elements in the program itself. Get the element to search for from the user. Print the position if element found.

Code:

```
#include <stdio.h>

int main() {
    int arr[] = {1, 2, 3, 5, 6, 7, 8};
    int n = sizeof(arr) / sizeof(arr[0]);
    int element, i;
    int found = -1;

    printf("Enter the element to search: ");
    scanf("%d", &element);

    for (i = 0; i < n; i++) {
        if (arr[i] == element) {
            printf("Element %d found at position %d.", element, i + 1);
            found = 1;
            break;
        }
    }

    if (found == -1) {
        printf("Element %d not found in the array.", element);
    }
    return 0;
}
```

Output:

```
C:\Users\Mclab\Desktop\21BE X + | - □ ×
Enter the element to search: 5
Element 5 found at position 4.
-----
Process exited after 2.401 seconds with return value 0
Press any key to continue . . . |
```

```
C:\Users\Mclab\Desktop\21BE X + | - □ ×
Enter the element to search: 20
Element 20 not found in the array.
-----
Process exited after 2.993 seconds with return value 0
Press any key to continue . . . |
```

2) Write a C program to find sum of all elements in an array.

Code:

```
#include<stdio.h>

int main(){

    int size, i, sum = 0;
    printf("Enter the size: ");
```

```

scanf("%d", &size);

int arr[size];

printf("Enter the elements: \n");

for (i = 0; i < size; i++){

    scanf("%d", &arr[i]);

}

for (i = 0; i < size; i++){

    sum += arr[i];

}

printf("The sum of the array is %d.", sum);

}

```

Output:

```

C:\Users\Mclab\Desktop\2IBE X + | v
Enter the size: 5
Enter the elements:
2 4 6 8 10
The sum of the array is 30.
Process exited after 5.507 seconds with return value 27
Press any key to continue . . .

```

- 3) Write a C program to find the smallest and largest elements in an array.

Code:

```

#include<stdio.h>

int main(){

    int size, i, smallest, largest;

```

```
printf("Enter the size: ");

scanf("%d", &size);

int arr[size];

printf("Enter the elements: \n");

for (i = 0; i < size; i++){

    scanf("%d", &arr[i]);

}

smallest = arr[0];

largest = arr[0];

for (i = 0; i < size; i++){

    if(arr[i] < smallest)

        smallest = arr[i];

    if(arr[i] > largest)

        largest = arr[i];

}

printf("The smallest element is %d and the largest element is %d.", smallest, largest);

return 0;

}
```

Output:

```
C:\Users\Mclab\Desktop\21BE + - X
Enter the size: 5
Enter the elements:
2 1 3 5 4
The smallest element is 1 and the largest element is 5.
Process exited after 16.51 seconds with return value 0
Press any key to continue . . . |
```

4) Write a C program to sort an array in ascending order.

Code:

```
#include<stdio.h>

int main(){

    int size, i, j, temp;

    printf("Enter the size: ");

    scanf("%d", &size);

    int arr[size];

    printf("Enter the elements: \n");

    for (i = 0; i < size; i++){

        scanf("%d", &arr[i]);

    }

    for (i = 0; i < (size-1); i++){

        for (j = 0; j < (size-i-1); j++){

            if (arr[j] > arr[j+1]){

                temp = arr[j];

                arr[j] = arr[j+1];

                arr[j+1] = temp;

            }

        }

    }

}
```

```

        arr[j+1] = temp;

    }

}

printf("The sorted array is: \n");

for (i = 0; i < size; i++){

    printf("%d ", arr[i]);

}

return 0;
}

```

**Output:**

```

C:\Users\Mclab\Desktop\21BE X + v
Enter the size: 5
Enter the elements:
6 3 8 2 4
The sorted array is:
2 3 4 6 8
-----
Process exited after 7.985 seconds with return value 0
Press any key to continue . . .

```

5) Write a C program to reverse the array elements.

**Code:**

```

#include<stdio.h>

int main(){

    int size, i, rev, temp;

    printf("Enter the size: ");

```

```
scanf("%d", &size);

int arr[size];

int new_arr[size];

printf("Enter the elements: \n");

for (i = 0; i < size; i++){

    scanf("%d", &arr[i]);

}

int start = 0;

int end = size - 1;

while (start < end) {

    temp = arr[start];

    arr[start] = arr[end];

    arr[end] = temp;

    start++;

    end--;

}

printf("Reversed array: \n");

for (i = 0; i < size; i++) {

    printf("%d ", arr[i]);

}

printf("\n");

return 0;

}
```

Output:

```
C:\Users\Mclab\Desktop\21BE + - X
Enter the size: 5
Enter the elements:
1 2 3 4 5
The reversed array is:
5 4 3 2 1
-----
Process exited after 4.132 seconds with return value 0
Press any key to continue . . . |
```

- 6) Write a C program for insertion of an element at a specified location.

Code:

```
#include<stdio.h>

int main(){

    int size, i, pos, num;

    printf("Enter the size: ");

    scanf("%d", &size);

    int arr[size];

    printf("Enter the elements: \n");

    for (i = 0; i < size; i++){

        scanf("%d", &arr[i]);

    }

    printf("Enter the number and position to insert: ");

    scanf("%d", &num);

    scanf("%d", &pos);

    size = size + 1;

    for(i = size-1; i>= pos; i--){
```

```

        arr[i] = arr[i-1];

    }

    arr[pos-1] = num;

    printf("The array after insertion is: \n");

    for (i = 0; i < size; i++){

        printf("%d ", arr[i]);

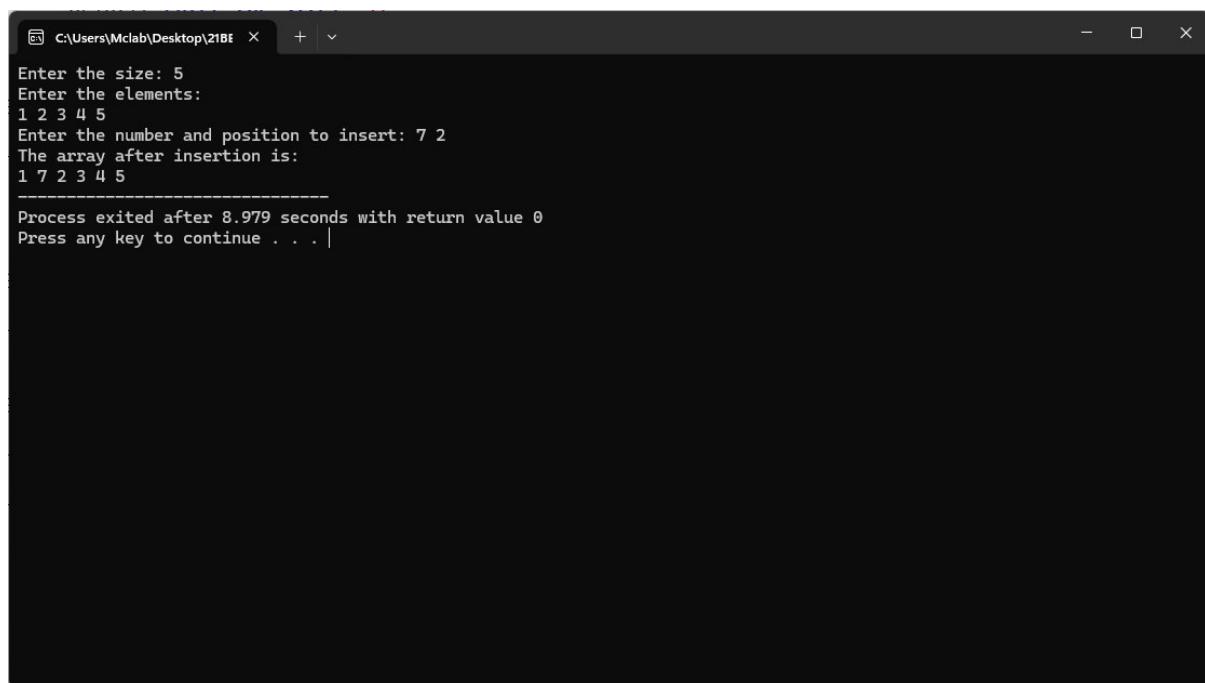
    }

    return 0;

}

```

Output:



```

C:\Users\MyLab\Desktop\2IBF X + v
Enter the size: 5
Enter the elements:
1 2 3 4 5
Enter the number and position to insert: 7 2
The array after insertion is:
1 7 2 3 4 5
-----
Process exited after 8.979 seconds with return value 0
Press any key to continue . . .

```

7) Write a C program to remove an element at a specified location.

Code:

```

#include <stdio.h>

int main() {

    int n, pos, i;

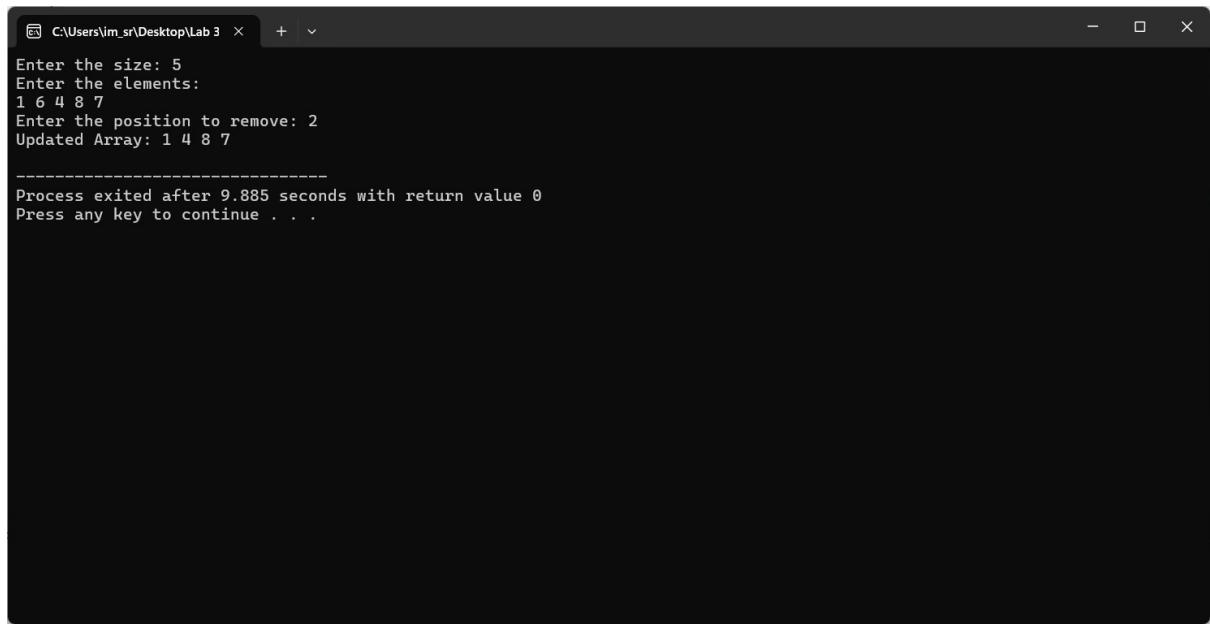
    printf("Enter the size: ");

    scanf("%d", &n);

```

```
int arr[n];
printf("Enter the elements:\n");
for(i = 0; i < n; i++) {
    scanf("%d", &arr[i]);
}
printf("Enter the position to remove: ");
scanf("%d", &pos);
if(pos < 0 || pos >= n){
    printf("Invalid position\n");
}
else{
    for(i = pos - 1; i < n - 1; i++) {
        arr[i] = arr[i + 1];
    }
    n--;
    printf("Updated Array: ");
    for(i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }
    printf("\n");
}
return 0;
}
```

Output:



```
C:\Users\im_sr\Desktop\Lab 3 + v
Enter the size: 5
Enter the elements:
1 6 4 8 7
Enter the position to remove: 2
Updated Array: 1 4 8 7
-----
Process exited after 9.885 seconds with return value 0
Press any key to continue . . .
```

8) Write a C program for addition of two matrices of any order.

Code:

```
#include <stdio.h>

int main() {

    int r, c;
    int i, j;
    printf("Enter the Number of Rows: ");
    scanf("%d", &r);
    printf("Enter the Number of Columns: ");
    scanf("%d", &c);

    int matrix1[r][c];
    int matrix2[r][c];
    int sumMatrix[r][c];

    printf("Enter the Elements of Matrix 1:\n");
    for(i = 0; i < r; i++) {
        for(j = 0; j < c; j++) {
            scanf("%d", &matrix1[i][j]);
        }
    }
```

```
}

printf("Enter the Elements of Matrix 2:\n");

for(i = 0; i < r; i++) {

    for(j = 0; j < c; j++) {

        scanf("%d", &matrix2[i][j]);

    }

}

for(i = 0; i < r; i++) {

    for(j = 0; j < c; j++) {

        sumMatrix[i][j] = matrix1[i][j] + matrix2[i][j];

    }

}

printf("The Sum of Two Matrix is:\n");

for(i = 0; i < r; i++) {

    for(j = 0; j < c; j++) {

        printf("%d ", sumMatrix[i][j]);

    }

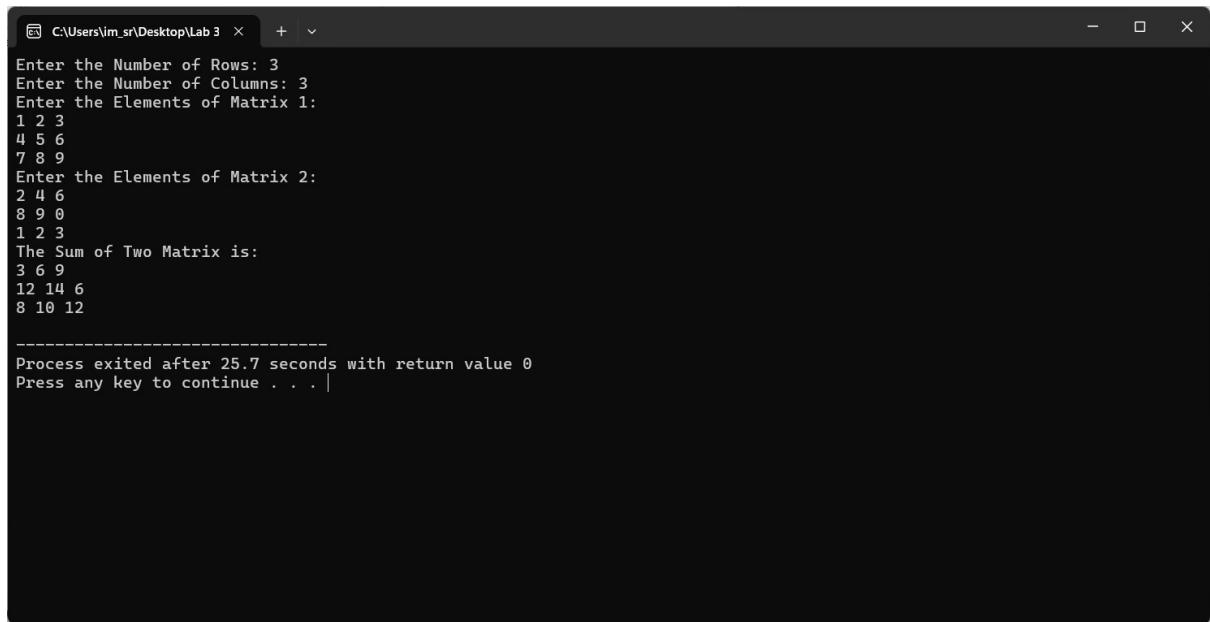
    printf("\n");

}

return 0;

}
```

Output:



```
C:\Users\im.sr\Desktop\Lab 3 + v
Enter the Number of Rows: 3
Enter the Number of Columns: 3
Enter the Elements of Matrix 1:
1 2 3
4 5 6
7 8 9
Enter the Elements of Matrix 2:
2 4 6
8 9 0
1 2 3
The Sum of Two Matrix is:
3 6 9
12 14 6
8 10 12
-----
Process exited after 25.7 seconds with return value 0
Press any key to continue . . . |
```

9) Write a C program to multiply two matrices.

Code:

```
#include <stdio.h>

int main() {

    int rows1, cols1, rows2, cols2;

    int i, j, k;

    printf("Enter the Number of Rows for Matrix 1: ");

    scanf("%d", &rows1);

    printf("Enter the Number of Columns for Matrix 1: ");

    scanf("%d", &cols1);

    printf("Enter the Number of Rows for Matrix 2: ");

    scanf("%d", &rows2);

    printf("Enter the Number of Columns for Matrix 2: ");

    scanf("%d", &cols2);

    if (cols1 != rows2) {

        printf("Matrices cannot be multiplied.\n");

        return 1;
    }
```

```

int matrix1[rows1][cols1];

int matrix2[rows2][cols2];

int productMatrix[rows1][cols2];

printf("Enter the elements of matrix 1:\n");

for(i = 0; i < rows1; i++) {

    for(j = 0; j < cols1; j++) {

        scanf("%d", &matrix1[i][j]);

    }

}

printf("Enter the elements of matrix 2:\n");

for(i = 0; i < rows2; i++) {

    for(j = 0; j < cols2; j++) {

        scanf("%d", &matrix2[i][j]);

    }

}

for(i = 0; i < rows1; i++) {

    for(j = 0; j < cols2; j++) {

        productMatrix[i][j] = 0;

        for(k = 0; k < cols1; k++) {

            productMatrix[i][j] += matrix1[i][k] * matrix2[k][j];

        }

    }

}

printf("The Multiplication of Two Matrix is:\n");

for(i = 0; i < rows1; i++) {

    for(j = 0; j < cols2; j++) {

        printf("%d ", productMatrix[i][j]);

    }

}

```

```

    }
    printf("\n");
}

return 0;
}

```

**Output:**

```

C:\Users\im_sr\Desktop\Lab 3 + v
Enter the Number of Rows for Matrix 1: 2
Enter the Number of Columns for Matrix 1: 2
Enter the Number of Rows for Matrix 2: 2
Enter the Number of Columns for Matrix 2: 2
Enter the elements of matrix 1:
1 2
3 4
Enter the elements of matrix 2:
5 6
7 8
The Multiplication of Two Matrix is:
19 22
43 50
-----
Process exited after 12.86 seconds with return value 0
Press any key to continue . . .

```

```

C:\Users\im_sr\Desktop\Lab 3 + v
Enter the Number of Rows for Matrix 1: 2
Enter the Number of Columns for Matrix 1: 1
Enter the Number of Rows for Matrix 2: 3
Enter the Number of Columns for Matrix 2: 2
Matrices cannot be multiplied.

-----
Process exited after 5.717 seconds with return value 1
Press any key to continue . . .

```

10) Write a C program to find the transpose of a matrix.

**Code:**

```
#include <stdio.h>

int main() {

    int r, c;

    int i, j;

    printf("Enter the Number of Rows: ");

    scanf("%d", &r);

    printf("Enter the Number of Columns: ");

    scanf("%d", &c);

    int matrix[r][c];

    int tran[c][r];

    printf("Enter the Elements:\n");

    for(i = 0; i < r; i++) {

        for(j = 0; j < c; j++) {

            scanf("%d", &matrix[i][j]);

        }

    }

    for(i = 0; i < r; i++) {

        for(j = 0; j < c; j++) {

            tran[j][i] = matrix[i][j];

        }

    }

    printf("The Transpose Matrix is:\n");

    for(i = 0; i < c; i++) {

        for(j = 0; j < r; j++) {

            printf("%d ", tran[i][j]);

        }

    }

    printf("\n");

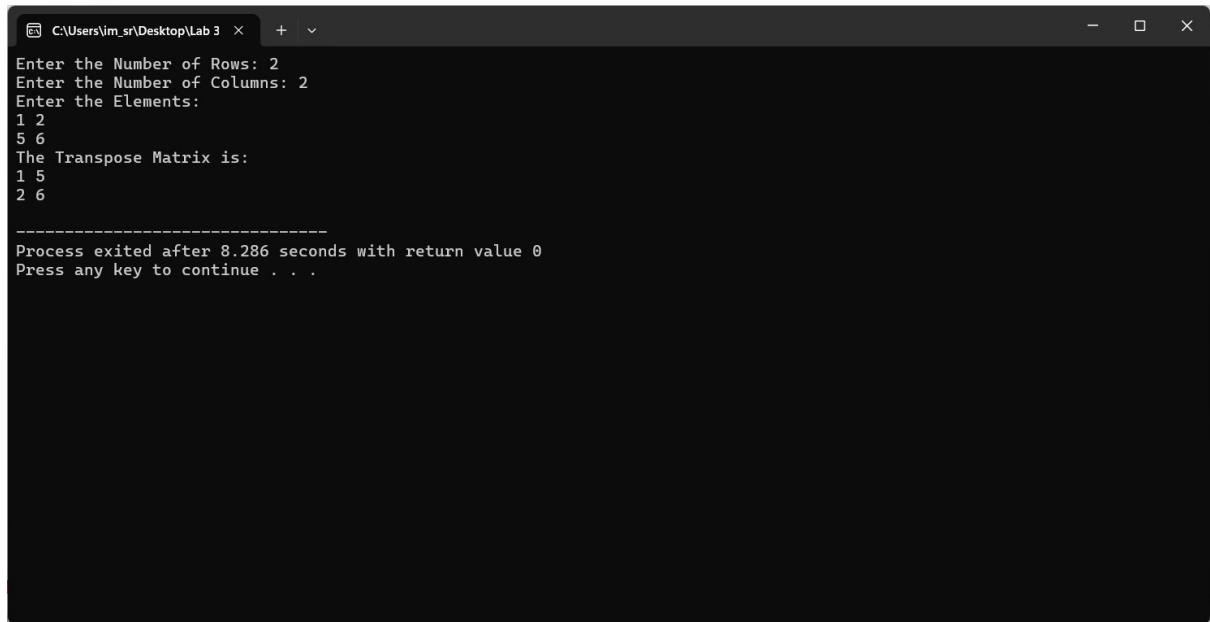
}
```

```
    }

    return 0;

}
```

Output:



```
C:\Users\im_sr\Desktop\Lab 3 > 
Enter the Number of Rows: 2
Enter the Number of Columns: 2
Enter the Elements:
1 2
5 6
The Transpose Matrix is:
1 5
2 6
-----
Process exited after 8.286 seconds with return value 0
Press any key to continue . . .
```

11) Write a C program to find average temperature of a week

Code:

```
#include <stdio.h>

int main() {
    float temp[7];

    float sum = 0, avg;

    int i;

    printf("Enter the Temperatures for the Week:\n");

    for(i = 0; i < 7; i++) {

        printf("Day %d: ", i + 1);

        scanf("%f", &temp[i]);

        sum += temp[i];
    }

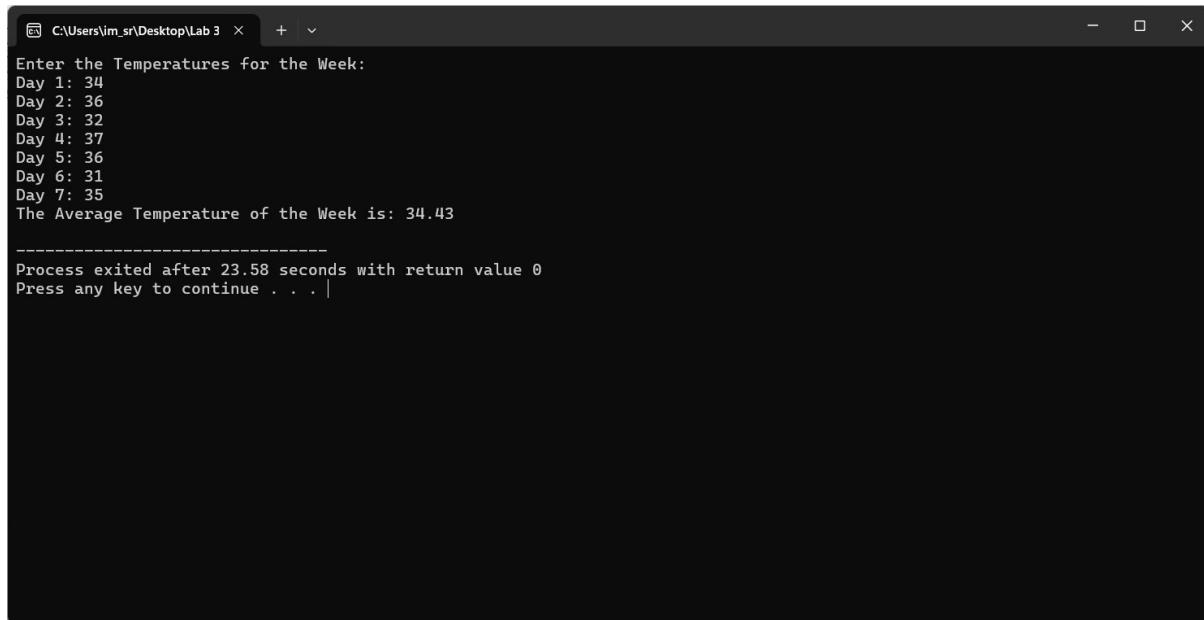
    avg = sum / 7;
```

```
    printf("The Average Temperature of the Week is: %.2f\n", avg);

    return 0;

}
```

Output:



```
C:\Users\im_sr\Desktop\Lab 3 > Enter the Temperatures for the Week:
Day 1: 34
Day 2: 36
Day 3: 32
Day 4: 37
Day 5: 36
Day 6: 31
Day 7: 35
The Average Temperature of the Week is: 34.43
-----
Process exited after 23.58 seconds with return value 0
Press any key to continue . . . |
```

**Optional:**

- 12) Develop an inventory management system using arrays in C. Get the product ids and no. of items from the user. Store them in two different arrays.

Code:

```
#include <stdio.h>

int main() {
    int n, i;

    printf("Enter the Number of Products: ");

    scanf("%d", &n);

    int prodid[n];
    int items[n];

    for(i = 0; i < n; i++) {
        printf("Enter the Product ID and Number of Items For Product %d: ", i + 1);
        scanf("%d %d", &prodid[i], &items[i]);
    }
}
```

```
}

printf("\nInventory Details:\n");

printf("Product ID\tNumber of Items\n");

for(i = 0; i < n; i++) {

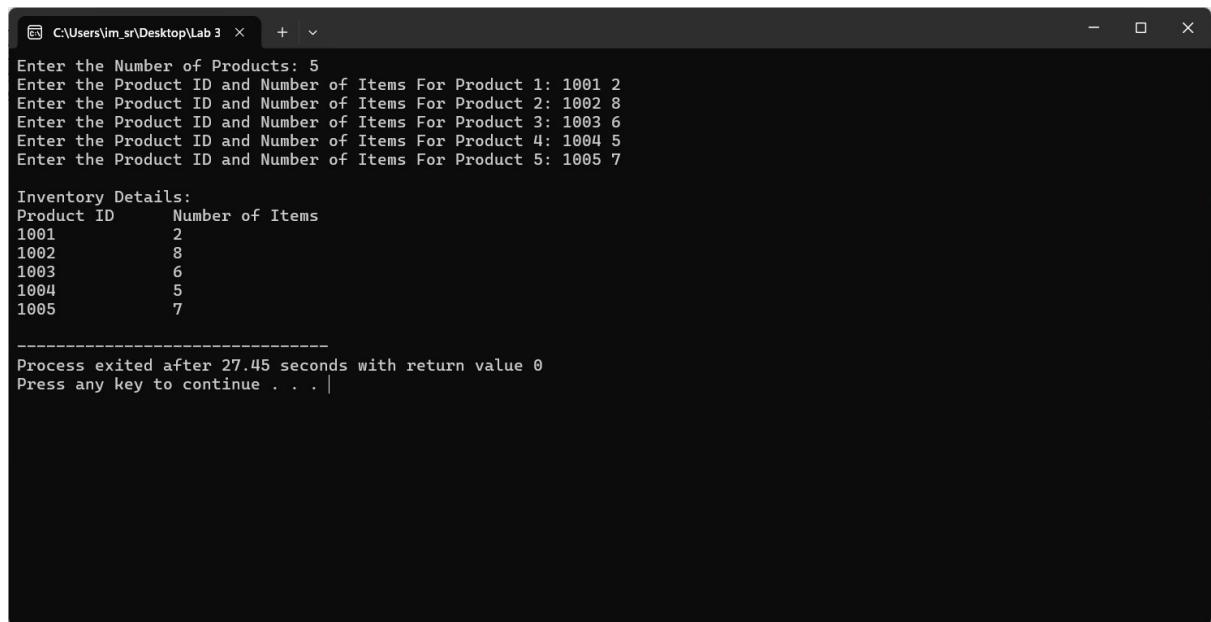
    printf("%d\t%d\n", prodid[i], items[i]);

}

return 0;

}
```

Output:



```
C:\Users\im_sr\Desktop\Lab 3 + v
Enter the Number of Products: 5
Enter the Product ID and Number of Items For Product 1: 1001 2
Enter the Product ID and Number of Items For Product 2: 1002 8
Enter the Product ID and Number of Items For Product 3: 1003 6
Enter the Product ID and Number of Items For Product 4: 1004 5
Enter the Product ID and Number of Items For Product 5: 1005 7

Inventory Details:
Product ID      Number of Items
1001            2
1002            8
1003            6
1004            5
1005            7

-----
Process exited after 27.45 seconds with return value 0
Press any key to continue . . . |
```

**Exercise 4:** Strings and Functions

**Register Number:** 21BEC1851

**Date:** 08-08-2024

- 1) Write a C program to count the number of vowels and consonants in the string

Code:

```
#include <stdio.h>
#include <string.h>
#include <ctype.h>

int main()
{
    char str[50];
    int i;
    printf("Enter a string: ");
    scanf("%s", str);
    int vowels = 0, consonants = 0;
    for (i = 0; str[i] != '\0'; i++) {
        char ch = tolower(str[i]);
        if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u') {
            vowels++;
        }
        else if (ch >= 'a' && ch <= 'z') {
            consonants++;
        }
    }
    printf("%d %d", vowels, consonants);
}
```

Output:

```
C:\Users\Mclab\Desktop\21BE + - X
Enter a string: Rahul
2 3
Process exited after 1.988 seconds with return value 3
Press any key to continue . . . |
```

2) Write a C program to check if the given string is palindrome.

Code:

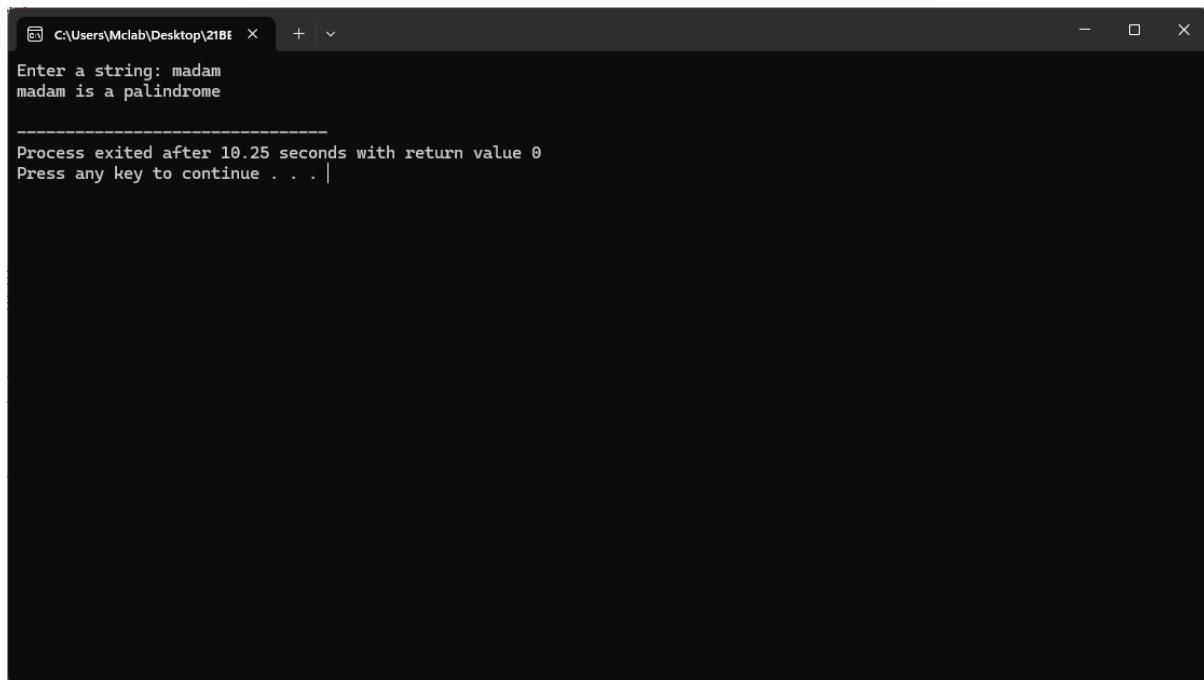
```
#include<stdio.h>
#include<string.h>
int main()
{
    char string1[20];
    printf("Enter a string: ");
    scanf("%s", string1);
    int l = 0;
    int h = strlen(string1) - 1;
    while (h > l) {
        if (string1[l++] != string1[h--]) {
            printf("%s is not a palindrome\n", string1);
            return 0;
        }
    }
}
```

```
    printf("%s is a palindrome\n", string1);

    return 0;

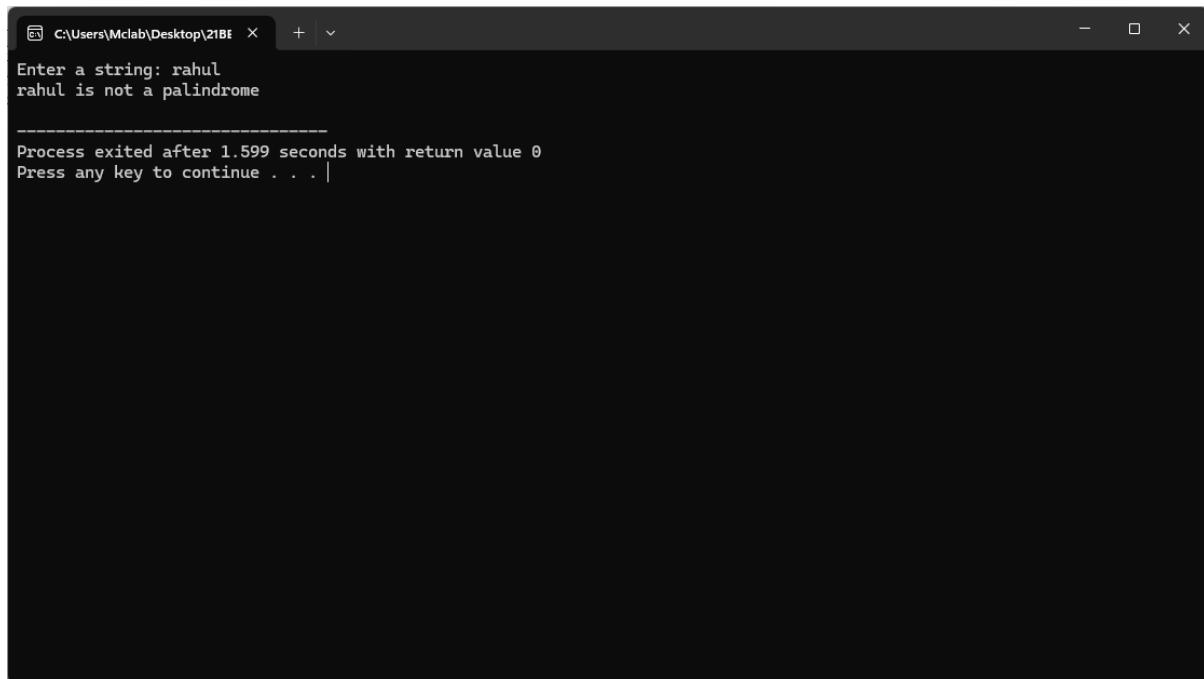
}
```

Output:



```
C:\Users\Mclab\Desktop\2IBF > Enter a string: madam
madam is a palindrome

-----
Process exited after 10.25 seconds with return value 0
Press any key to continue . . . |
```



```
C:\Users\Mclab\Desktop\2IBF > Enter a string: rahul
rahul is not a palindrome

-----
Process exited after 1.599 seconds with return value 0
Press any key to continue . . . |
```

- 3) Write a C program to remove spaces from a given string

Code:

```
#include<stdio.h>
```

```
#include<string.h>

int main()
{
    char str[100];

    int i, j;

    printf("Enter a string: ");

    fgets(str,100,stdin);

    int h = strlen(str);

    for (i = 0; i<h;i++) {

        if(str[i] == ' '){

            for (j = i; j<h; j++)

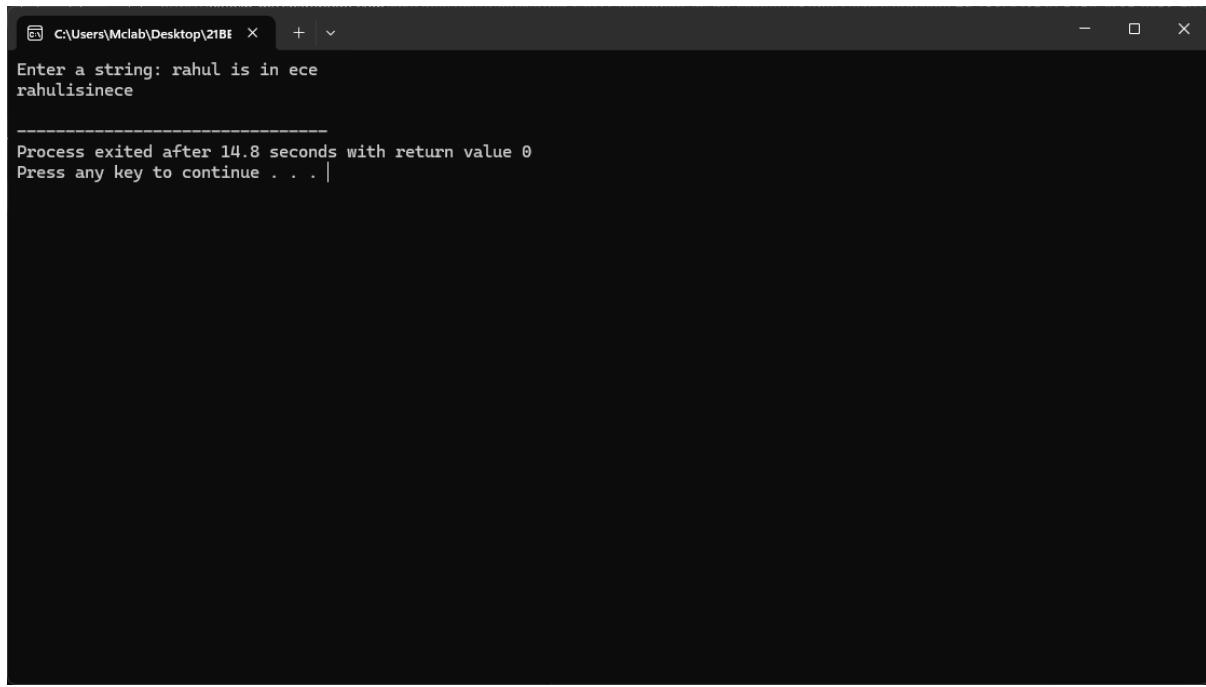
                str[j]=str[j+1];

            i--;
        }
    }

    printf("%s",str);

    return 0;
}
```

Output:



```
C:\Users\Mclab\Desktop\21BE X + v
Enter a string: rahul is in ece
rahul is in ece

-----
Process exited after 14.8 seconds with return value 0
Press any key to continue . . . |
```

- 4) Write a C program to read two strings and compare them using the function strcmp() and print a message that the first string is equal, less, or greater than the second one

Code:

```
#include <stdio.h>
#include <string.h>

int main() {
    char str1[100], str2[100];
    printf("Enter the first string: ");
    fgets(str1, sizeof(str1), stdin);
    printf("Enter the second string: ");
    fgets(str2, sizeof(str2), stdin);

    int result = strcmp(str1, str2);

    if (result == 0) {
        printf("The strings are equal.\n");
    } else
        if (result < 0) {
            printf("The first string is less than the second string.\n");
        } else {
```

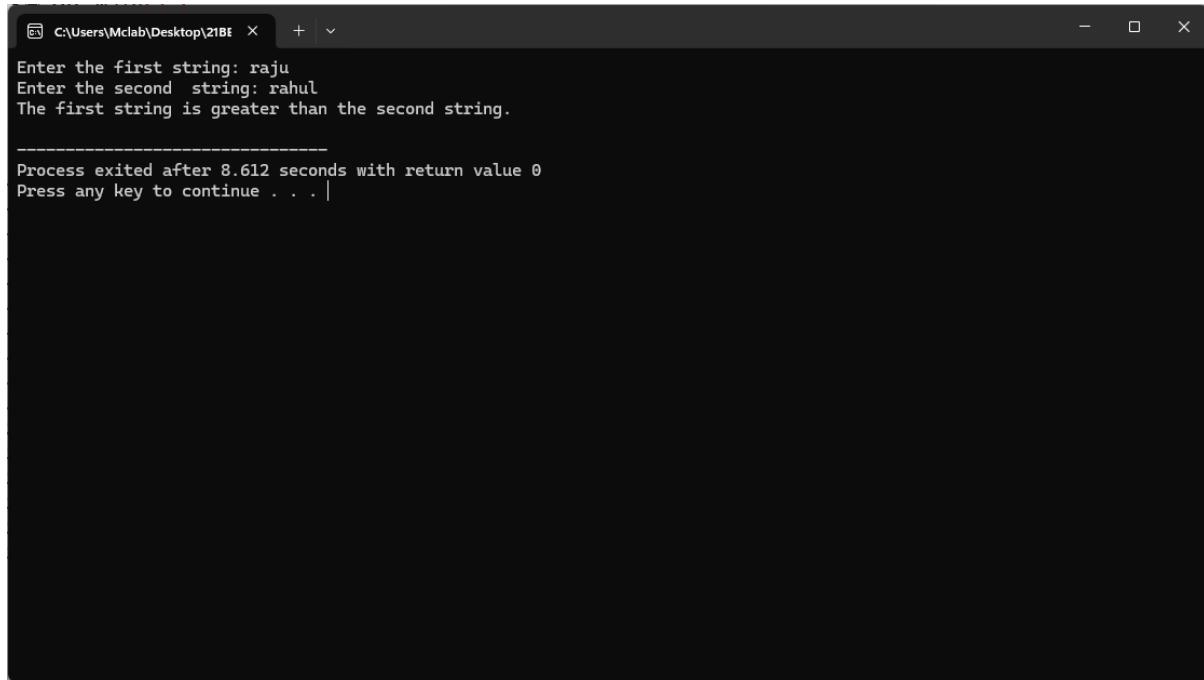
```
    printf("The first string is greater than the second string.\n");

}

return 0;

}
```

Output:



```
C:\Users\Mclab\Desktop\21BCE + - X
Enter the first string: raju
Enter the second string: rahul
The first string is greater than the second string.

-----
Process exited after 8.612 seconds with return value 0
Press any key to continue . . . |
```

- 5) Design a user name and password authentication system using C programming

Code:

```
#include <stdio.h>

#include <string.h>

int main() {

char usr[] = "rahul";

char pass[] = "helloworld";

char str1[100], str2[100];

printf("Enter the username: ");

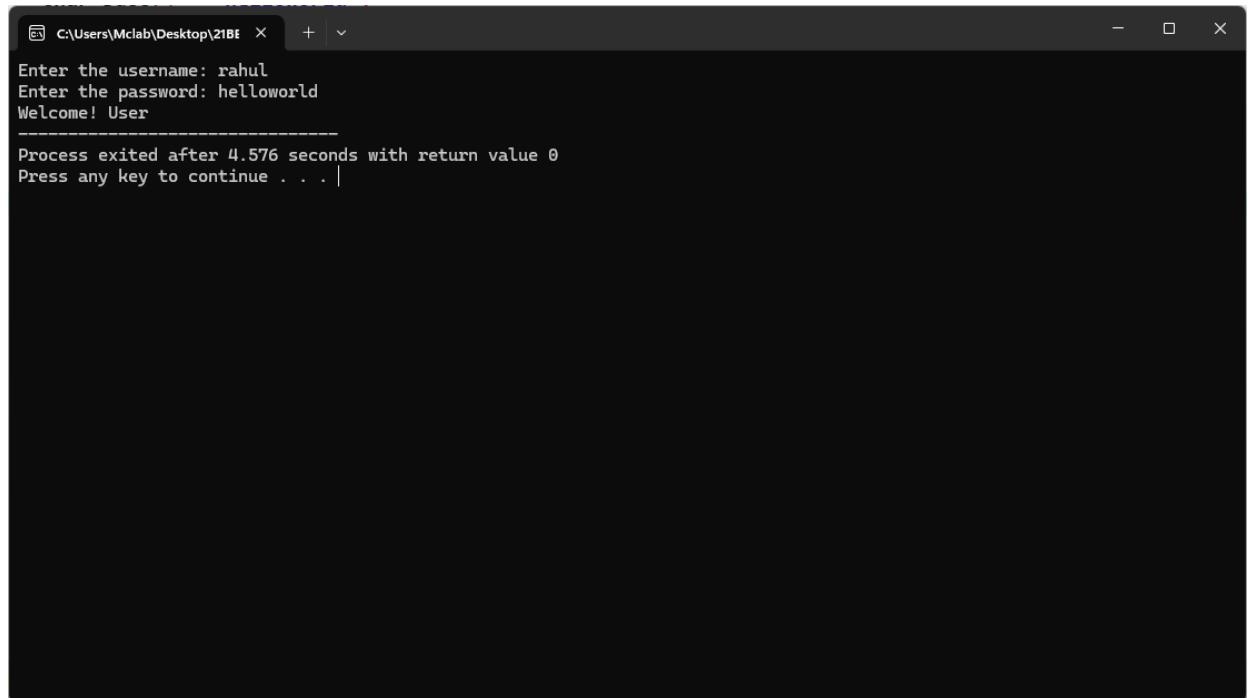
scanf("%s", str1);

printf("Enter the password: ");

scanf("%s", str2);
```

```
int usrres = strcmp(str1, usr);
int passres = strcmp(str2, pass);
if ((usrres == 0) && (passres == 0)) {
    printf("Welcome! User");
} else{
    printf("Access denied");
}
return 0;
}
```

Output:



```
C:\Users\Mclab\Desktop\Z1BE X
Enter the username: rahul
Enter the password: helloworld
Welcome! User
-----
Process exited after 4.576 seconds with return value 0
Press any key to continue . . . |
```

The screenshot shows a terminal window with the following text:

```
C:\Users\Mclab\Desktop\21BE X + - □ ×  
Enter the username: rahul  
Enter the password: password  
Access denied  
-----  
Process exited after 7.189 seconds with return value 0  
Press any key to continue . . . |
```

- 6) Write a C program to encode a given string by adding number 5 to the ASCII value of its characters and print the encoded string

Code:

```
#include <stdio.h>  
  
#include <string.h>  
  
int main() {  
  
    char str1[100], ch;  
  
    int i;  
  
    printf("Enter the String: ");  
  
    fgets(str1, 100, stdin);  
  
    for (i = 0; str1[i] != '\0'; i++) {  
  
        str1[i] = str1[i] + 5;  
  
    }  
  
    printf("The Encoded String: %s", str1);  
  
    return 0;  
}
```

Output:

```
C:\Users\im.sr\Desktop\Lab 4 > Enter the String: rahul
The Encoded String: wfmzq
-----
Process exited after 8.623 seconds with return value 0
Press any key to continue . . . |
```

- 7) Write a C program with a function prime that returns 1 if its argument is a prime number and returns zero otherwise.

Code:

```
#include<stdio.h>

int isPrime(int num){

    int i;
    if (num <= 1)
        return 0;
    for (i = 2; i < num; i++){
        if (num % i == 0){
            return 0;
        }
    }
    return 1;
}

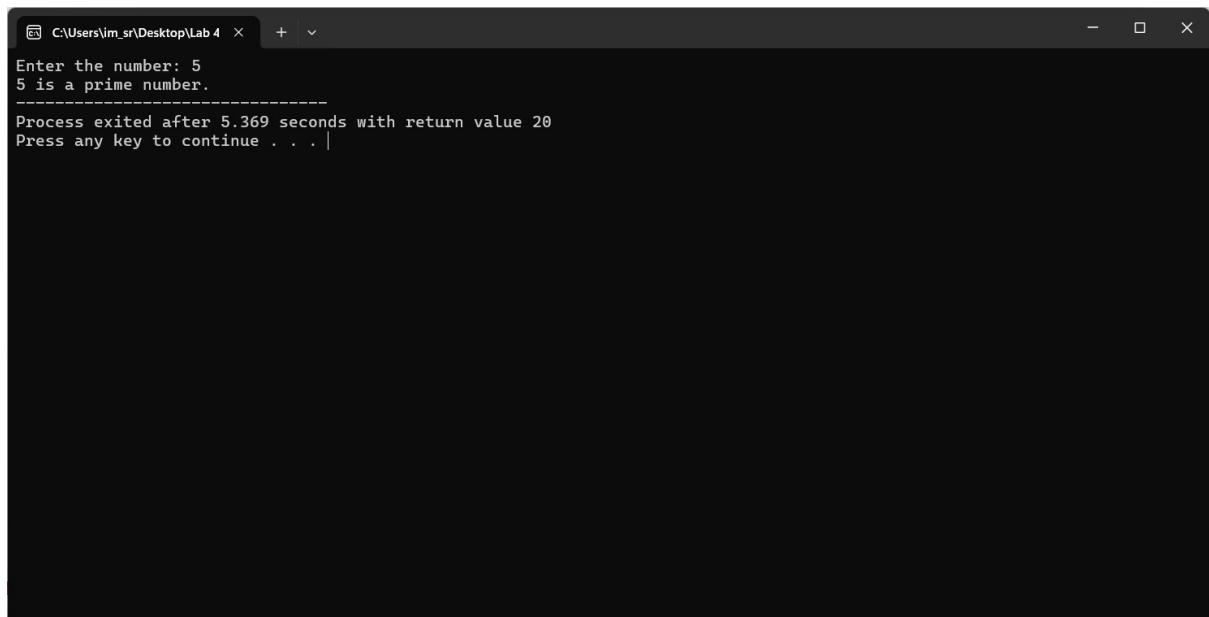
int main(){

    int number;
```

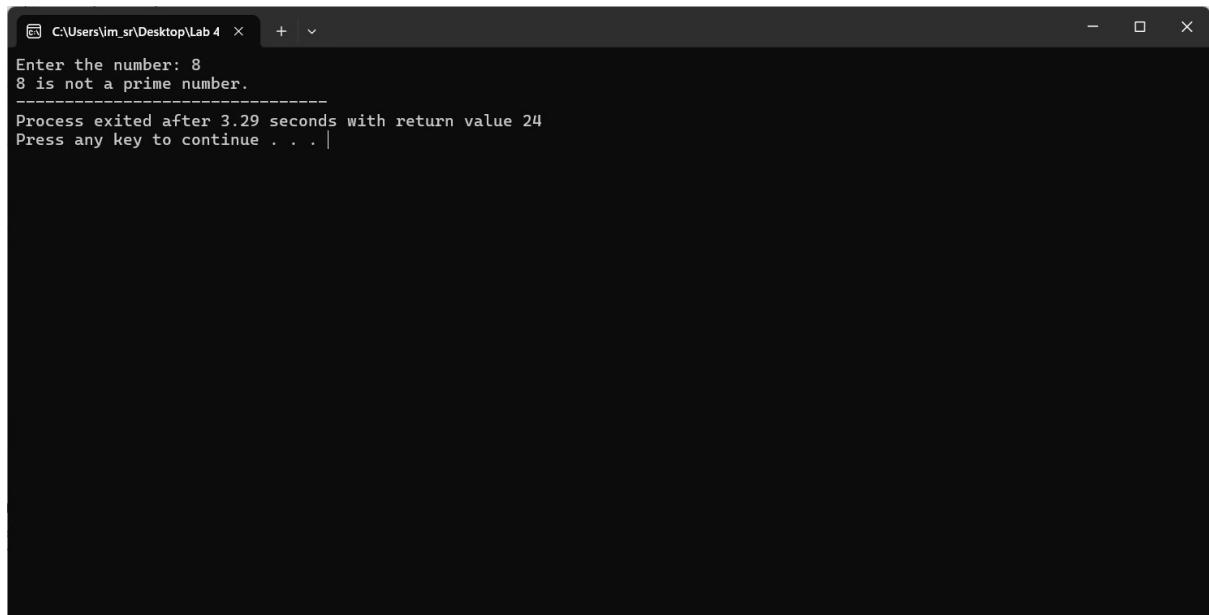
```
printf("Enter the number: ");
scanf("%d", &number);
int check = isPrime(number);
if (check == 1) printf("%d is a prime number.", number);
else printf("%d is not a prime number.", number);

}
```

Output:



```
C:\Users\im_sr\Desktop\Lab 4 >
Enter the number: 5
5 is a prime number.
-----
Process exited after 5.369 seconds with return value 20
Press any key to continue . . . |
```



```
C:\Users\im_sr\Desktop\Lab 4 >
Enter the number: 8
8 is not a prime number.
-----
Process exited after 3.29 seconds with return value 24
Press any key to continue . . . |
```

- 8) Write a C program to find the factorial of a number using recursive function

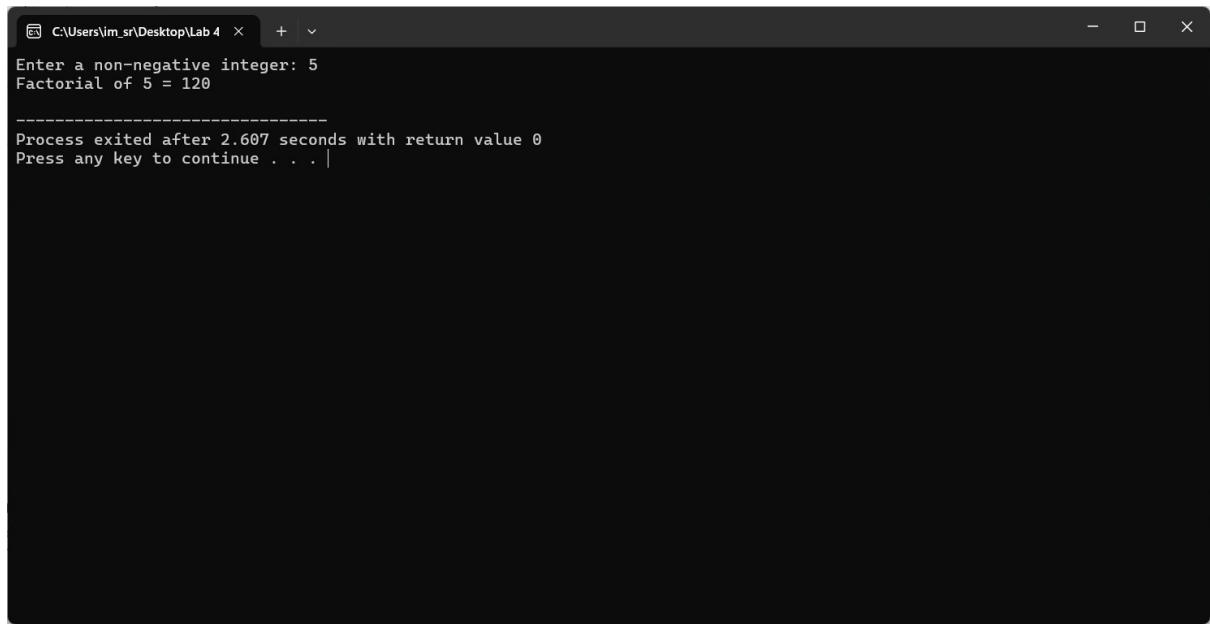
Code:

```
#include <stdio.h>

int factorial(int n) {
    if (n == 0) {
        return 1;
    }
    else {
        return n * factorial(n - 1);
    }
}

int main() {
    int num;
    printf("Enter a non-negative integer: ");
    scanf("%d", &num);
    printf("Factorial of %d = %d\n", num, factorial(num));
    return 0;
}
```

Output:



```
C:\Users\im.sr\Desktop\Lab 4 > Factorial of 5 = 120
-----
Process exited after 2.607 seconds with return value 0
Press any key to continue . . . |
```

9) Write a C program for nPr and nCr using functions

Code:

```
#include <stdio.h>

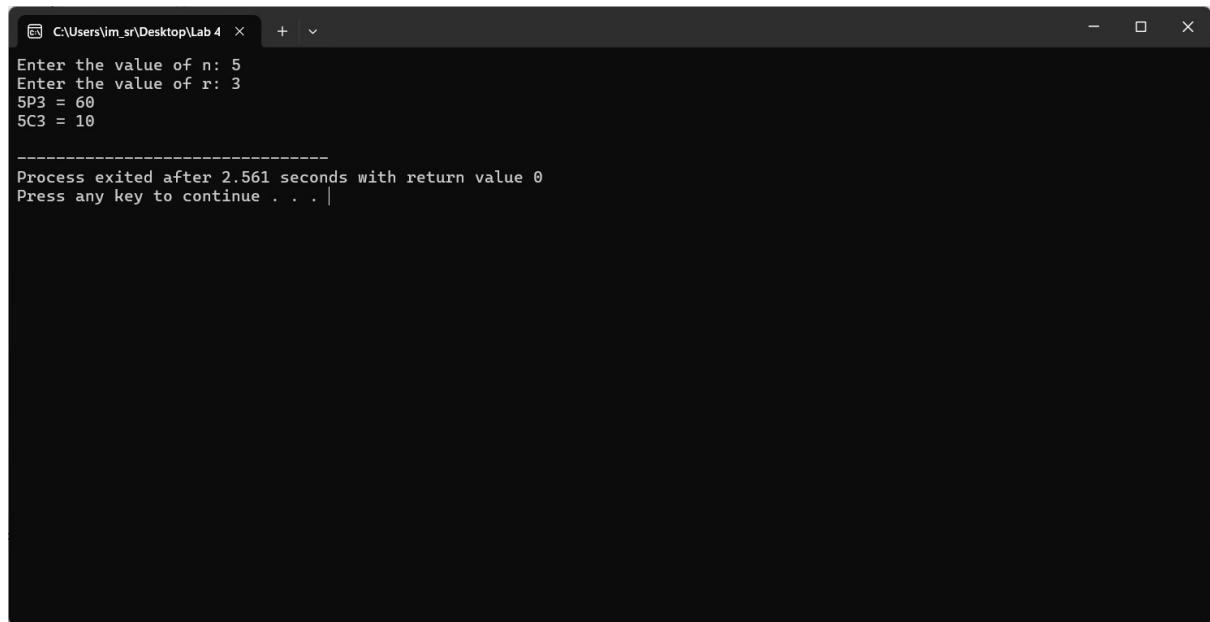
long factorial(int n) {
    if (n == 0 || n == 1)
        return 1;
    else
        return n * factorial(n - 1);
}

long nPr(int n, int r) {
    return factorial(n) / factorial(n - r);
}

long nCr(int n, int r) {
    return factorial(n) / (factorial(r) * factorial(n - r));
}
```

```
int main() {
    int n, r;
    printf("Enter the value of n: ");
    scanf("%d", &n);
    printf("Enter the value of r: ");
    scanf("%d", &r);
    printf("%dP%d = %ld\n", n, r, nPr(n, r));
    printf("%dC%d = %ld\n", n, r, nCr(n, r));
    return 0;
}
```

Output:



```
C:\Users\im_sr\Desktop\Lab 4
Enter the value of n: 5
Enter the value of r: 3
5P3 = 60
5C3 = 10
-----
Process exited after 2.561 seconds with return value 0
Press any key to continue . . . |
```

- 10) Write a function that will generate and print the first n Fibonacci numbers using recursion.

Code:

```
#include <stdio.h>

int Fibonacci(int n) {
    if (n == 1 || n == 2)
        return 1;
```

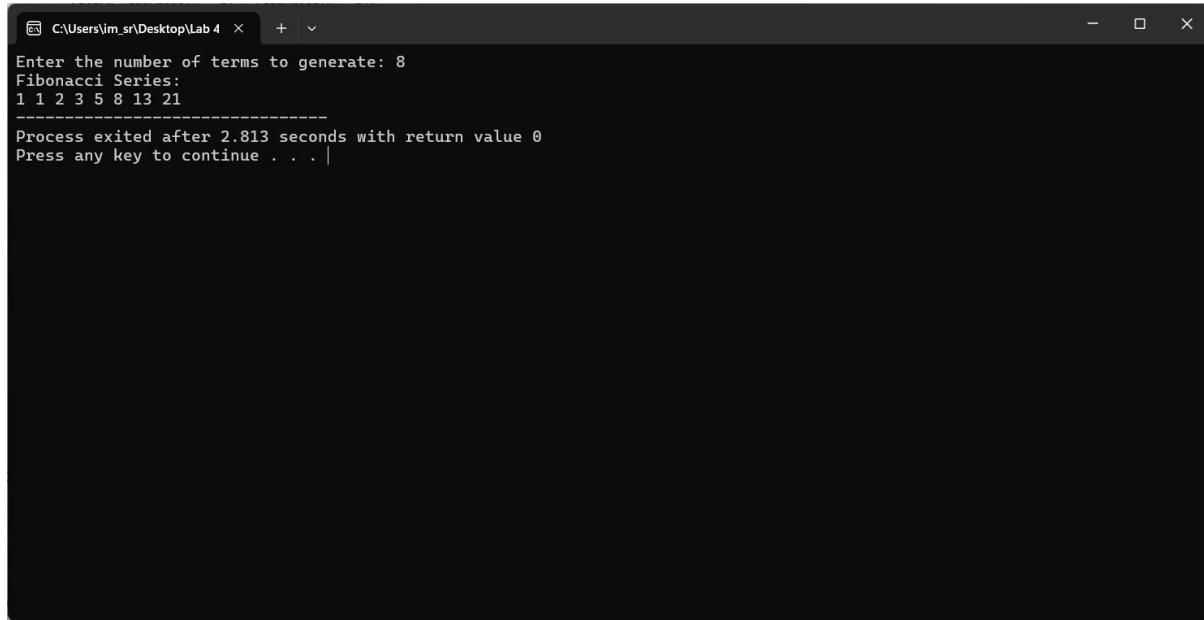
```

    else
        return Fibonacci(n - 1) + Fibonacci(n - 2);
    }

int main() {
    int n, i;
    printf("Enter the number of terms to generate: ");
    scanf("%d", &n);
    printf("Fibonacci Series: \n");
    for (i = 1; i <= n; i++) {
        printf("%d ", Fibonacci(i));
    }
    return 0;
}

```

Output:



```

C:\Users\im_sr\Desktop\Lab 4
Enter the number of terms to generate: 8
Fibonacci Series:
1 1 2 3 5 8 13 21
Process exited after 2.813 seconds with return value 0
Press any key to continue . . .

```

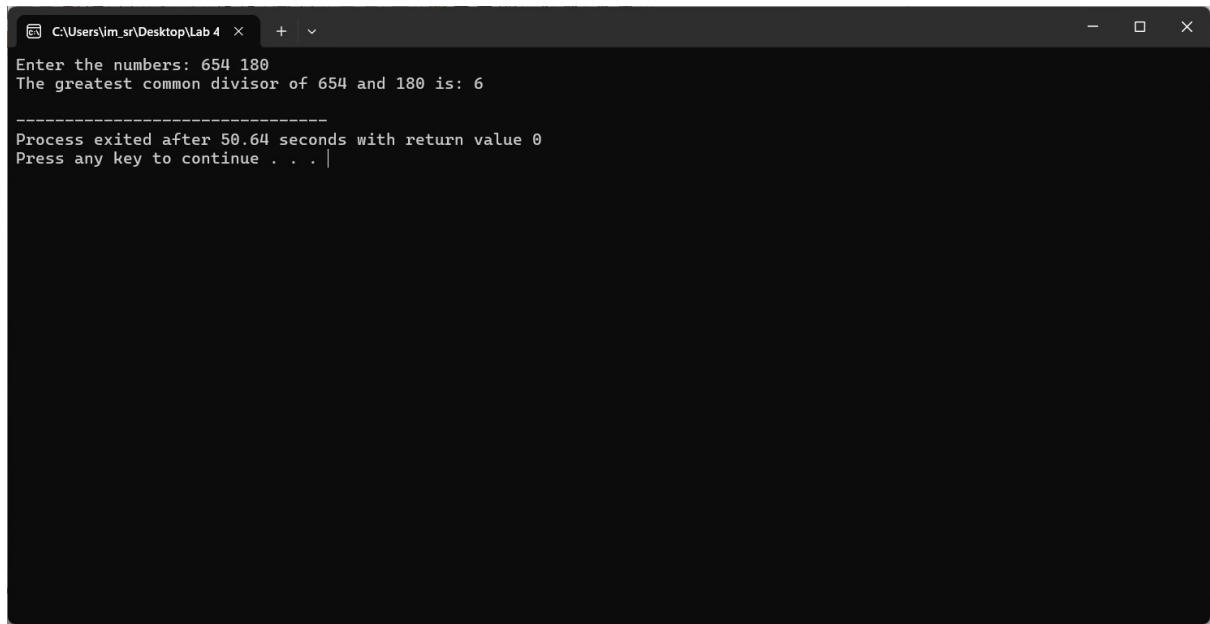
12) Write a function to compute the greatest common divisor given by Euclid's algorithm.

Code:

```
#include <stdio.h>
```

```
int gcd(int a, int b) {  
    while (b != 0) {  
        int div, c;  
        div = a / b;  
        c = a - div*b;  
        a = b;  
        if (c == 0)  
            return b;  
        b = c;  
    }  
}  
  
int main() {  
    int J, K;  
    printf("Enter the numbers: ");  
    scanf("%d %d", &J, &K);  
    printf("The greatest common divisor of %d and %d is: %d\n", J, K, gcd(J, K));  
    return 0;  
}
```

Output:



```
C:\Users\im.sr\Desktop\Lab 4 > Enter the numbers: 654 180
The greatest common divisor of 654 and 180 is: 6
-----
Process exited after 50.64 seconds with return value 0
Press any key to continue . . . |
```

13) Write a recursive function to obtain the running sum of first n natural numbers.

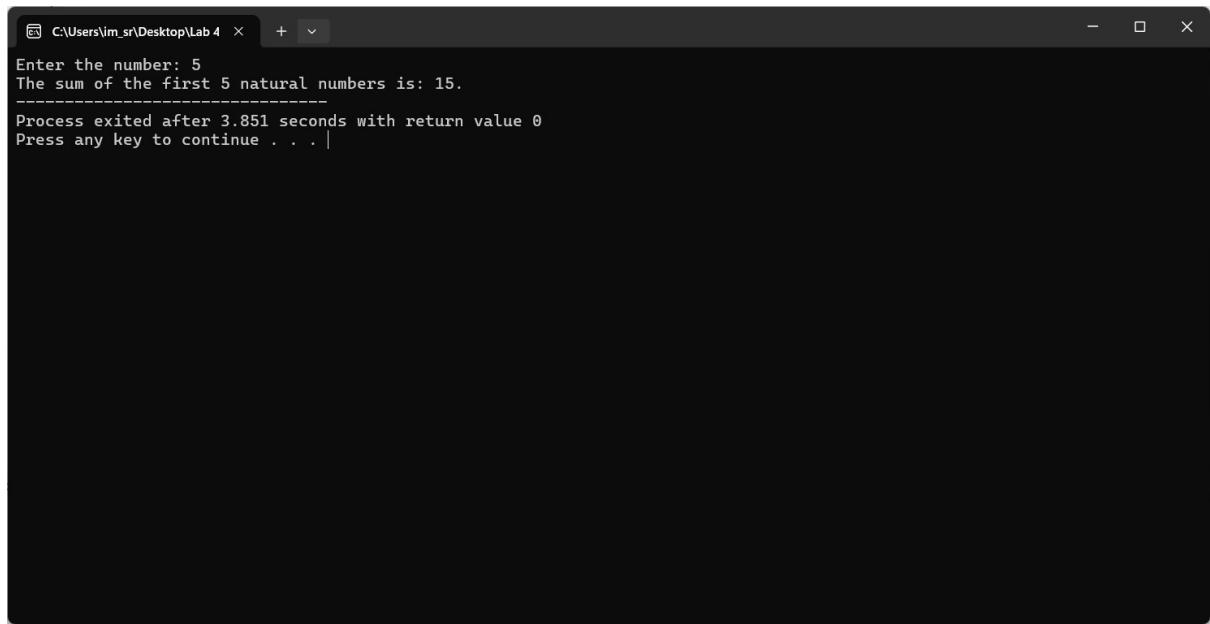
Code:

```
#include <stdio.h>

int usrsum(int n) {
    if (n == 1)
        return 1;
    else
        return n + usrsum(n - 1);
}

int main() {
    int n;
    printf("Enter the number: ");
    scanf("%d", &n);
    printf("The sum of the first %d natural numbers is: %d.", n, usrsum(n));
    return 0;
}
```

Output:



- 14) Write a program that extracts part of the given string from the specified position. If the number of characters to be extracted is 0 then the program should extract entire string from the specified position.

Code:

```
#include <stdio.h>

int main() {
    char source[50];
    int position, no_of_char;
    printf("Enter the string:\n");
    fgets(source, 50, stdin);
    printf("Enter the Position and Number of Characters: ");
    scanf("%d %d", &position, &no_of_char);
    char result[50];
    int i = 0;
    if (no_of_char == 0) {
        while (source[position-1] != '\0') {
            result[i] = source[position-1];
            i++;
        }
        position++;
    }
}
```

```
    }

} else {

    while (i < no_of_char && source[position-1] != '\0') {

        result[i] = source[position-1];

        i++;

        position++;

    }

}

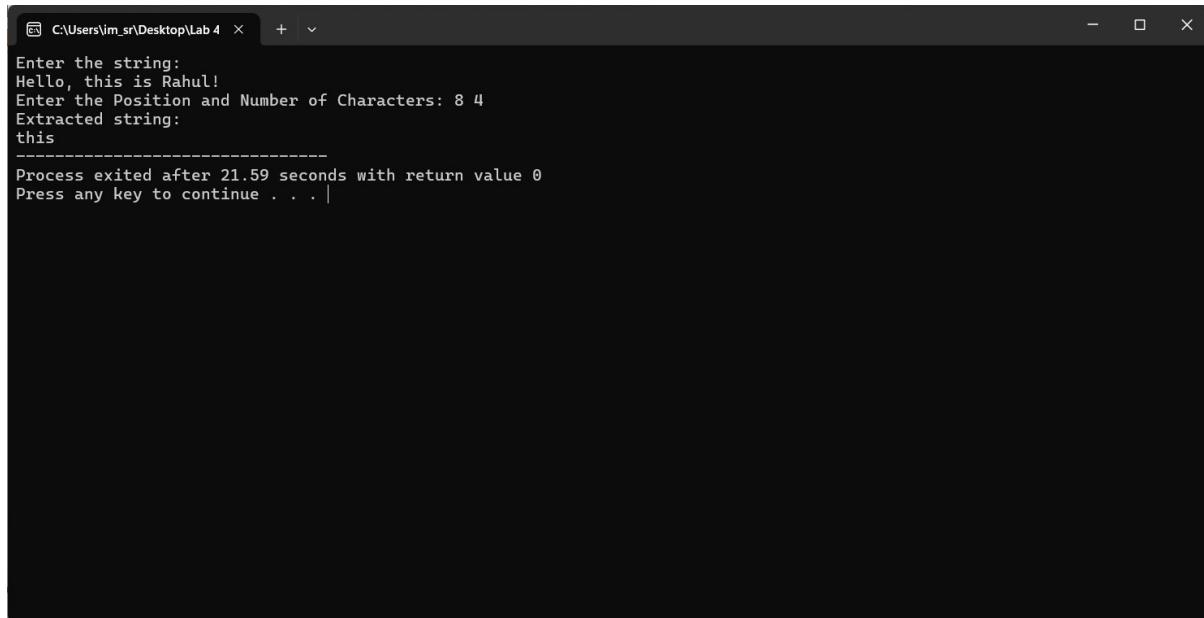
result[i] = '\0';

printf("Extracted string: \n%s", result);

return 0;

}
```

Output:



```
C:\Users\im_sr\Desktop\Lab 4 + ▾
Enter the string:
Hello, this is Rahul!
Enter the Position and Number of Characters: 8 4
Extracted string:
this
-----
Process exited after 21.59 seconds with return value 0
Press any key to continue . . . |
```

```
C:\Users\im.sr\Desktop\Lab 4 + - X
Enter the string:
Hello, this is Rahul!
Enter the Position and Number of Characters: 8 0
Extracted string:
this is Rahul!

-----
Process exited after 18.87 seconds with return value 0
Press any key to continue . . . |
```

## Exercise 5 and 6: Programming with 8051 I/O Ports, Timers and Counters

**Register Number:** 21BEC1851

**Date:** 22-08-2024

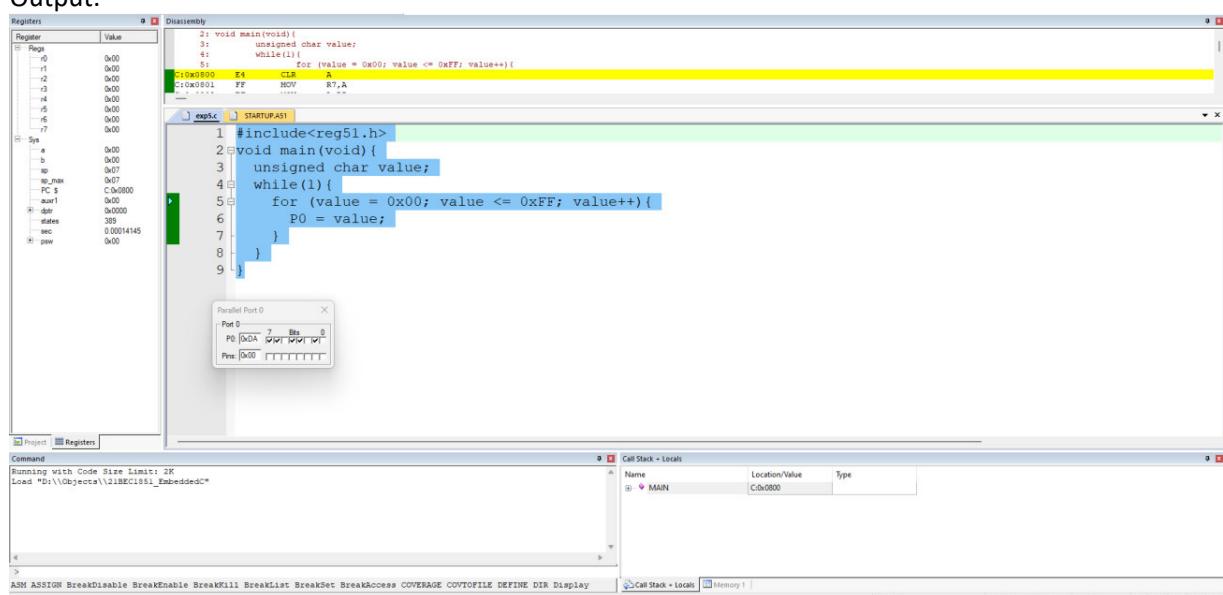
### **Programming with 8051 I/O Ports**

- 1) Write an embedded C program to send values 00 – FF to port P0 of 8051.

Code:

```
#include<reg51.h>
void main(void){
    unsigned char value;
    while(1){
        for (value = 0x00; value <= 0xFF; value++){
            P0 = value;
        }
    }
}
```

Output:



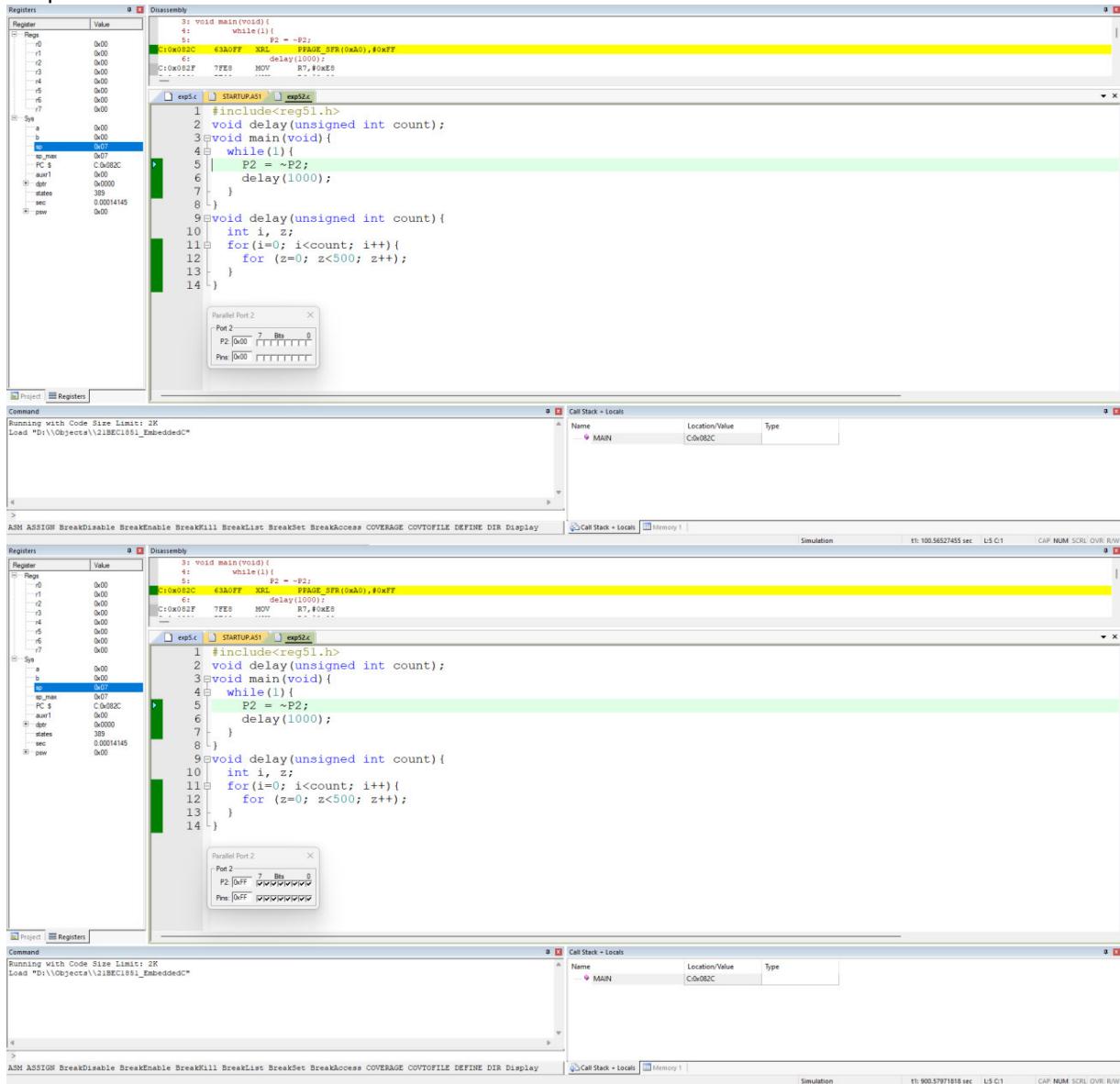
- 2) Write an embedded C program to toggle all bits of P2 continuously.

Code:

```
#include<reg51.h>
void delay(unsigned int count);
void main(void){
    while(1{
        P2 = ~P2;
        delay(1000);
    }
}
void delay(unsigned int count){
    int i, z;
    for(i=0; i<count; i++){
        for (z=0; z<500; z++);
    }
}
```

```
}
```

**Output:**



- 3) Write an embedded C program to toggle bit D0 of port P1 (P1.0) 50,000 times. Verify it in hardware.

Code:

```
#include<reg51.h>
sbit pin1 = P1^0;
void delay(unsigned int count);
void main(void){
unsigned int d = 0;
while(d <= 50000){
pin1 = ~pin1;
```

```

delay(1000);
d++;
}
while(1);
}

void delay(unsigned int count){
int i, z;
for(i=0; i<count; i++){
for (z=0; z<500; z++);
}
}

```

### Output:

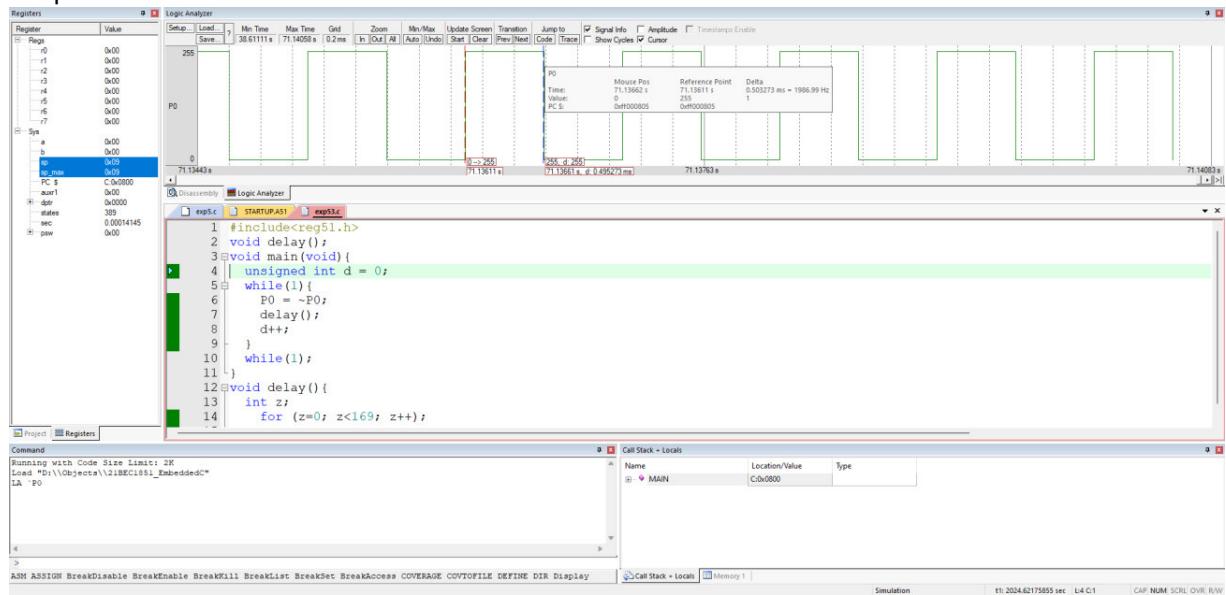
The screenshot displays a debugger environment with multiple windows. The top window is the 'Registers' window, showing CPU registers (r0-r7, SREG, PC, PSW) and memory locations. The bottom window is the 'Memory' window, showing memory dump and hex dump. Between these are two main code windows: 'Disassembly' and 'Source'. The 'Disassembly' window shows assembly code with highlighted instructions, while the 'Source' window shows the corresponding C code. A call stack and locals window is also visible.

- 4) Write an embedded C program to toggle bits of P0 port continuously with a 500 ms delay. Verify it in hardware.

Code:

```
#include<reg51.h>
void delay();
void main(void){
unsigned int d = 0;
while(1){
P0 = ~P0;
delay();
d++;
}
while(1);
}
void delay(){
int z;
for (z=0; z<169; z++);
}
```

Output:



- 5) Write an embedded C program to configure port 0 as an input port. This port is read continuously and assigned to P1 if the value is less than 100 else the value is output on P2.

Code:

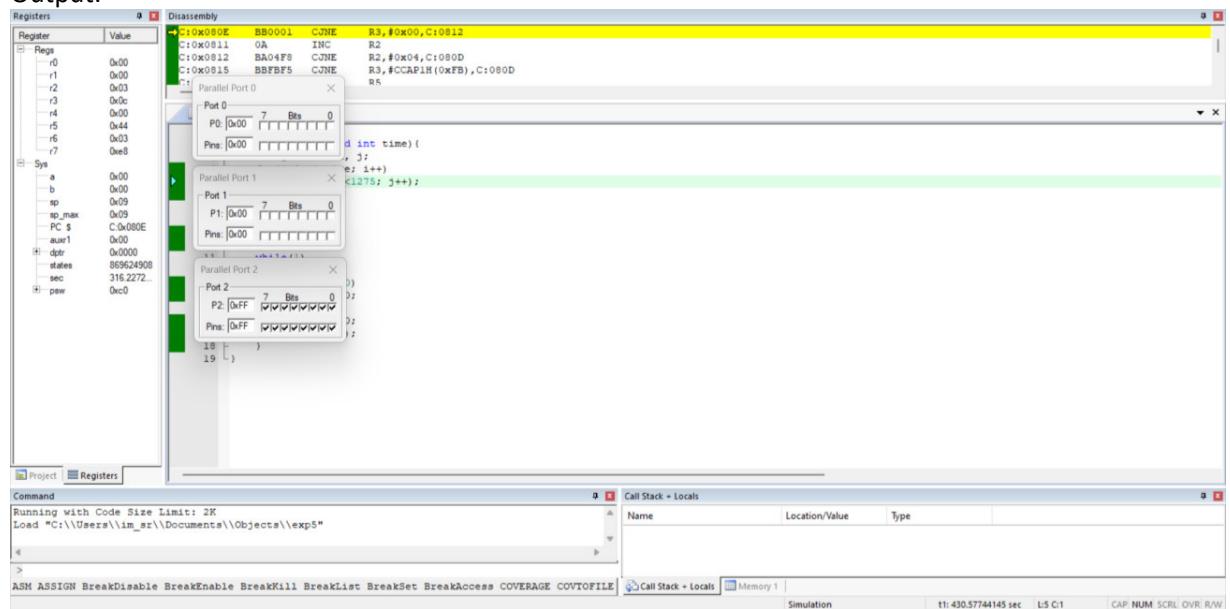
```
#include <reg51.h>
void delay(unsigned int time){
unsigned int i, j;
for(i=0; i<time; i++)
    for(j=0; j<1275; j++);
}
```

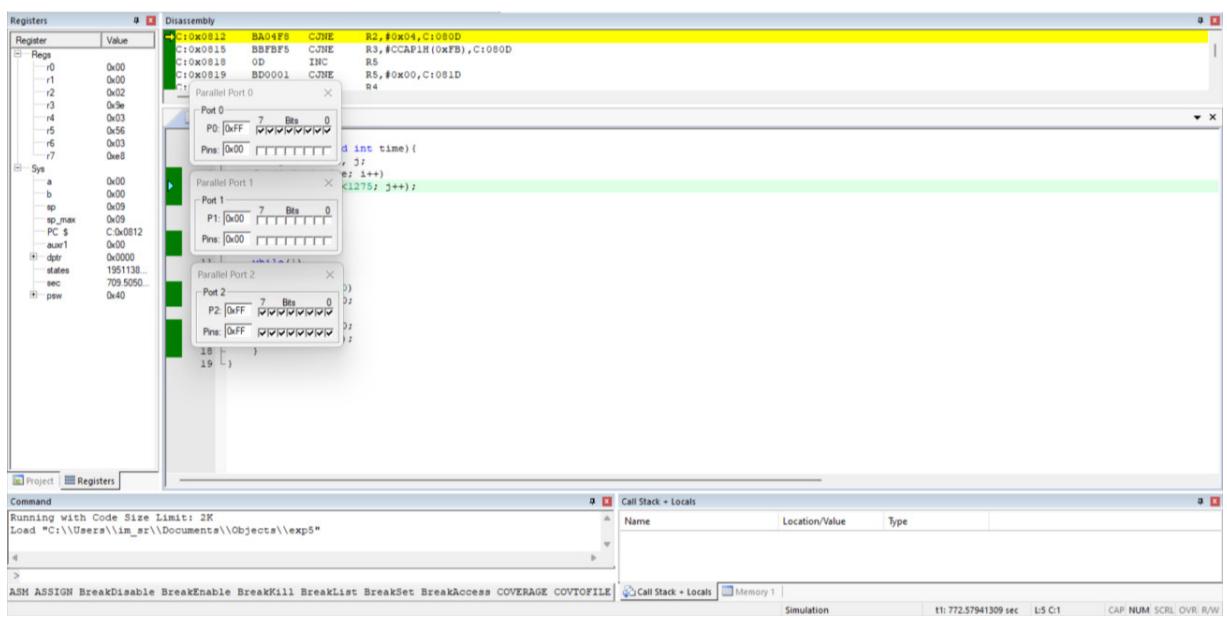
```

void main(){
    P0 = 0xFF;
    P2 = 0x00;
    while(1)
    {
        if(P0 < 100)
            P1 = P0;
        else
            P2 = P0;
        delay(1000);
    }
}

```

### Output:



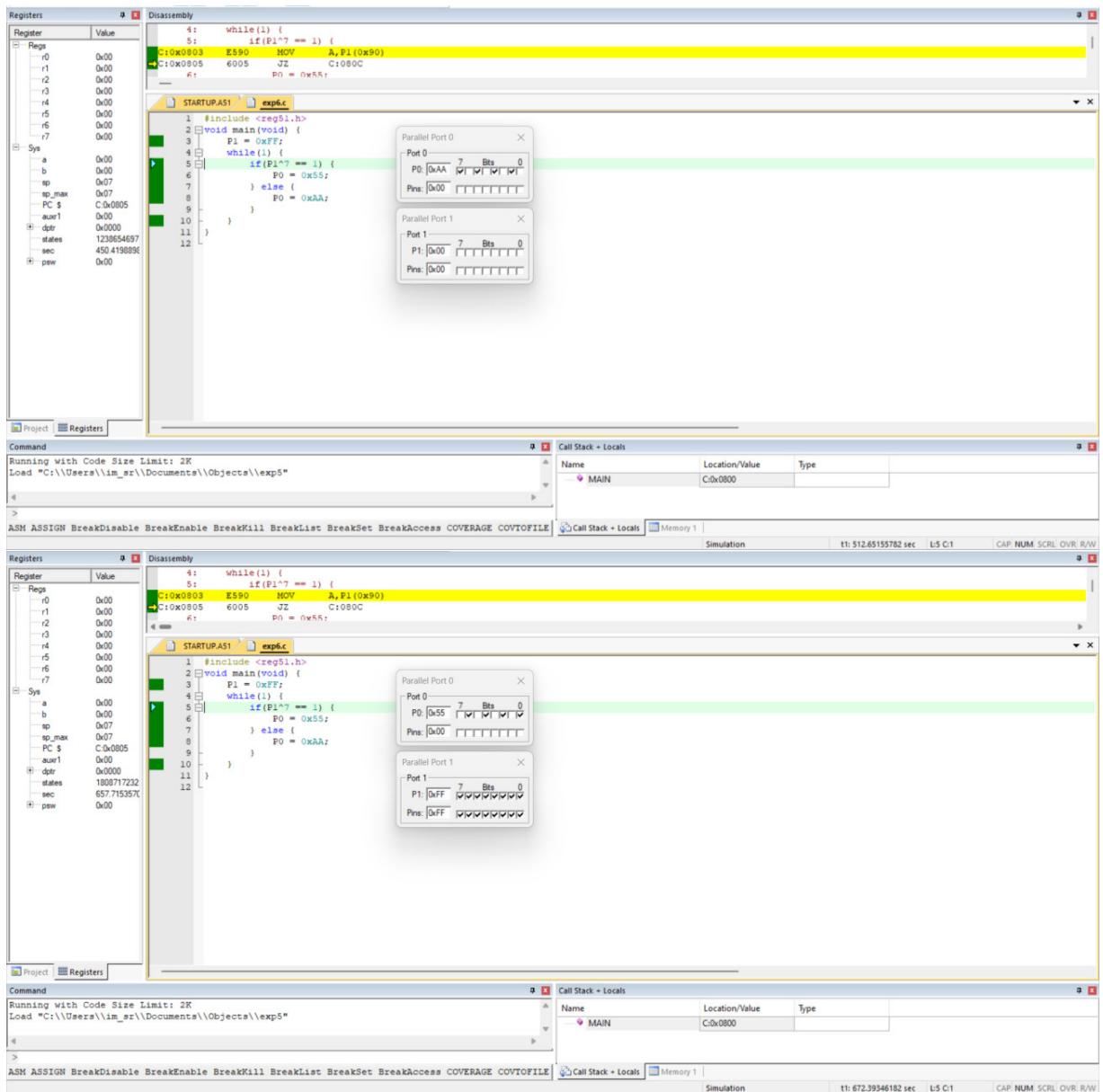


6) Write embedded C program to put 0x55 on P0 if the P1.7 status is 1, else to put 0xAA.

Code:

```
#include <reg51.h>
void main(void) {
    P1 = 0xFF;
    while(1) {
        if(P1^7 == 1) {
            P0 = 0x55;
        } else {
            P0 = 0xAA;
        }
    }
}
```

Output:



7) Implement binary counter in P3 using embedded C program.

Code:

```
#include <reg51.h>
void delay(unsigned int time)
{
    unsigned int i, j;
    for(i=0; i<time; i++)
        for(j=0; j<1275; j++);
}

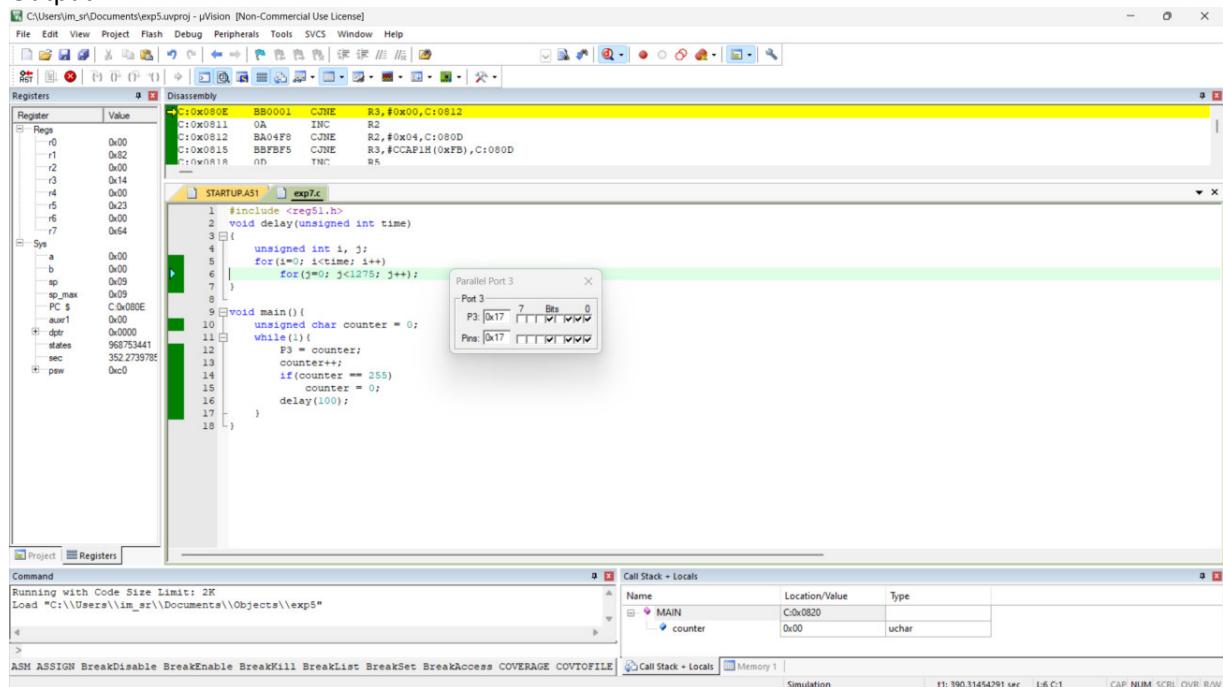
void main(){
    unsigned char counter = 0;
```

```

while(1){
    P3 = counter;
    counter++;
    if(counter == 255)
        counter = 0;
    delay(100);
}

```

### Output:



## Timers

- 8) Write an 8051 embedded C program to toggle port P1 continuously with 10 msec delay. Use timer 0 in 16-bit mode for delay.

Code:

```

#include <reg51.h>
#define TIMER0_RELOAD_HIGH 0xDC
#define TIMER0_RELOAD_LOW 0x00
void delay_10ms(void) {
    // Load Timer 0 registers
    TH0 = 0xDC;
    TL0 = 0x00;

    TR0 = 1;
    while (TF0 == 0);
    TR0 = 0;
    TF0 = 0;
}

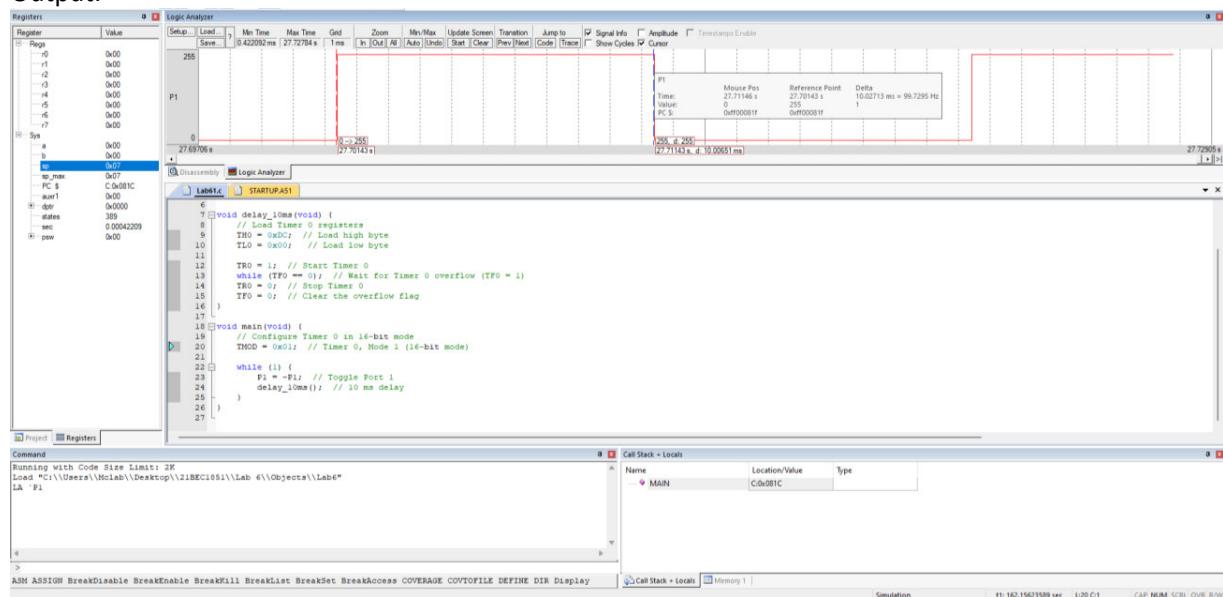
```

```

void main(void) {
    TMOD = 0x01;
    while (1) {
        P1 = ~P1;
        delay_10ms();
    }
}

```

### Output:



- 9) Write an 8051 embedded C program to generate a square wave of 500 Hz frequency on pin P2.3. Use timer 1 in mode 1.

Code:

```

#include <reg51.h>
sbit out=P2^3;
// Define constants
#define TIMER0_RELOAD_HIGH 0xDC
#define TIMER0_RELOAD_LOW 0x00

```

```

void delay_1ms(void) {
    TH1 = 0xFC;
    TL1 = 0x66;

    TR1 = 1;
    while (TF1 == 0);
    TR1 = 0;
    TF1 = 0;
}

void main(void) {
    TMOD = 0x10;
}

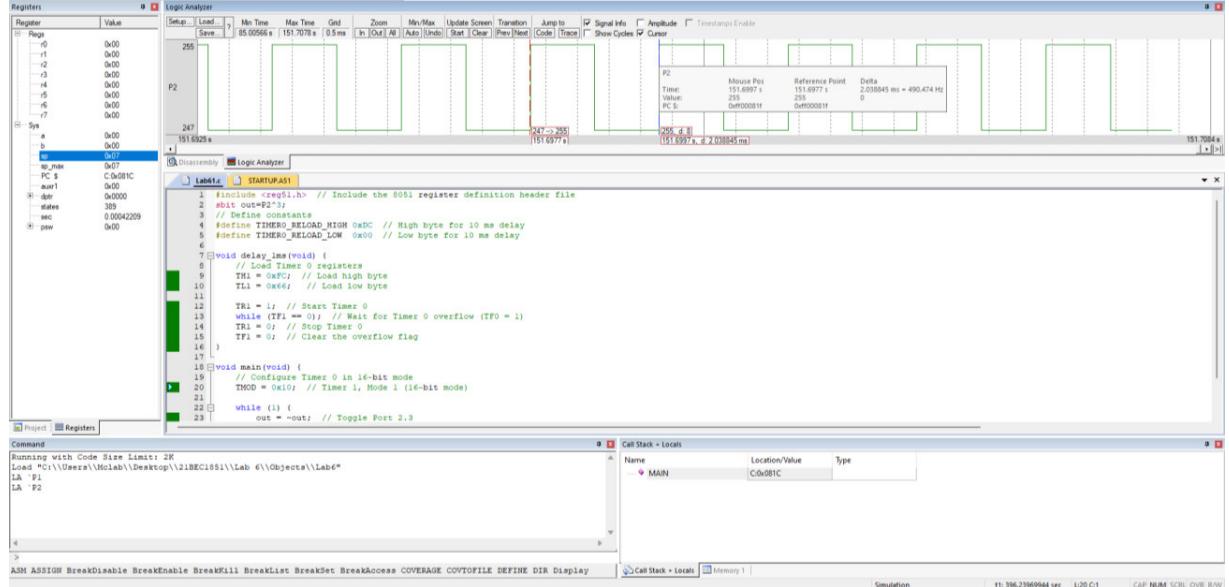
```

```

while (1) {
    out = ~out;
    delay_1ms();
}

```

Output:



- 10) Write an 8051 embedded C program to create a frequency of 2500Hz on pin P2.7. Use timer 1, mode 2 to create the delay.

Code:

```

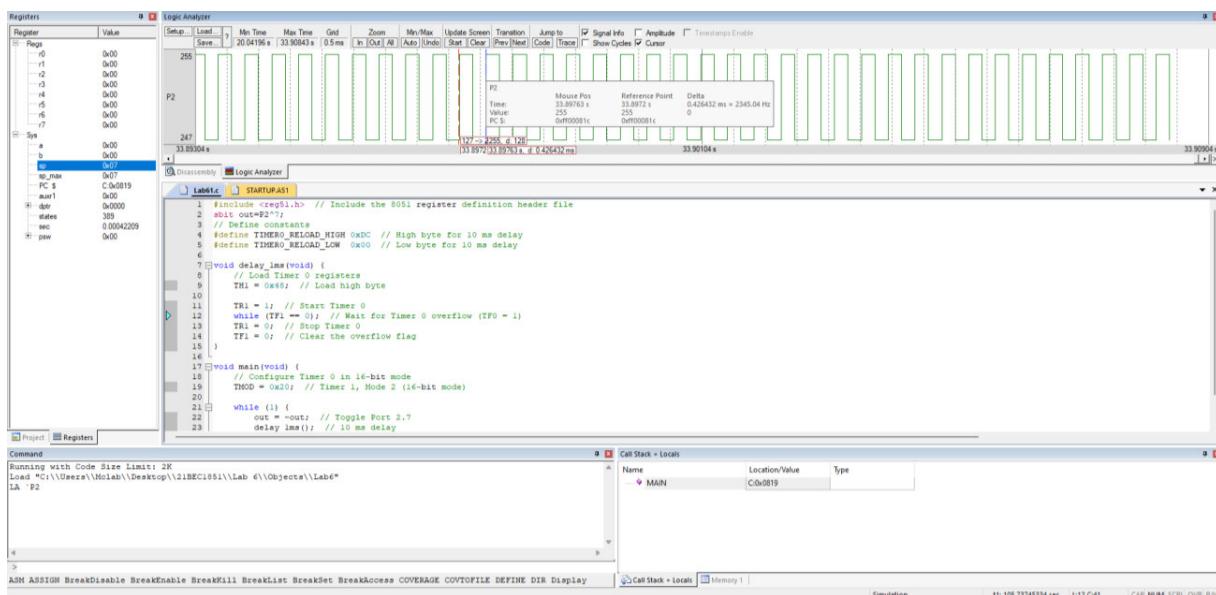
#include <reg51.h>
sbit out=P2^7;
#define TIMERO_RELOAD_HIGH 0xDC
#define TIMERO_RELOAD_LOW 0x00
void delay_1ms(void) {
    // Load Timer 0 registers
    TH1 = 0x48;

    TR1 = 1;
    while (TF1 == 0);
    TR1 = 0;
    TF1 = 0;
}

void main(void) {
    TMOD = 0x20;
    while (1) {
        out = ~out;
        delay_1ms();
    }
}

```

Output:



11) Write a program to generate a square wave of 2 kHz frequency on pin P1.5.

Code:

```
#include <reg51.h>
sbit out=P1^5;
#define TIMER0_RELOAD_HIGH 0xDC
#define TIMER0_RELOAD_LOW 0x00
```

```
void delay_1ms(void) {
```

```
    TH1 = 0xFF;
    TL1 = 0x19;
```

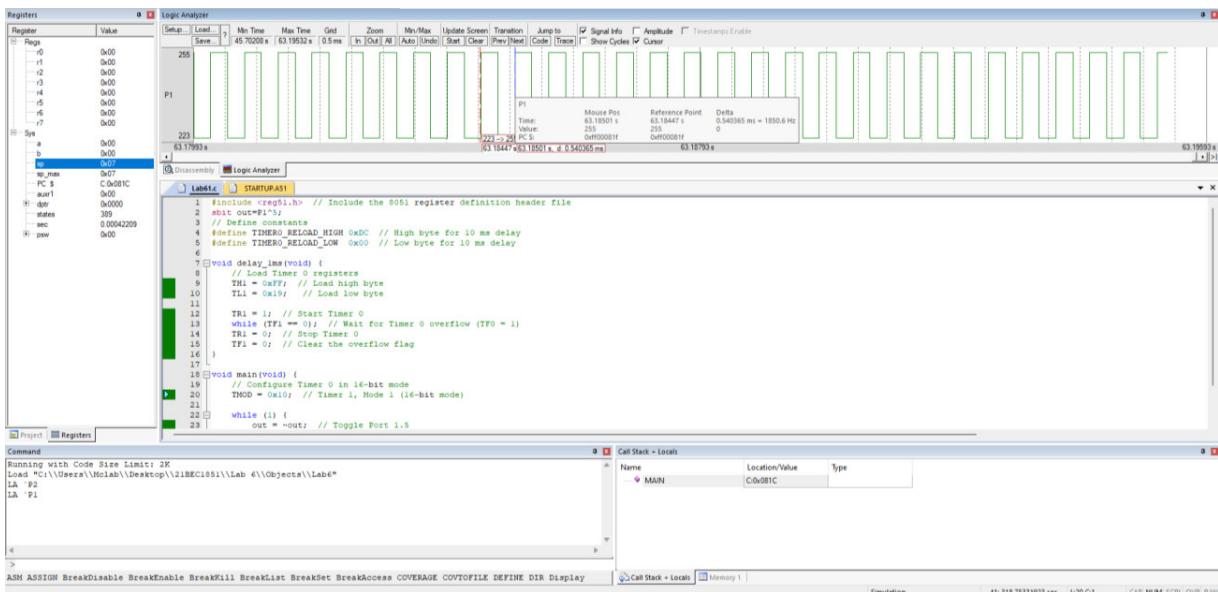
```
    TR1 = 1;
    while (TF1 == 0);
    TR1 = 0;
    TF1 = 0;
}
```

```
void main(void) {
```

```
    TMOD = 0x10;
```

```
    while (1) {
        out = ~out;
        delay_1ms();
    }
}
```

Output:

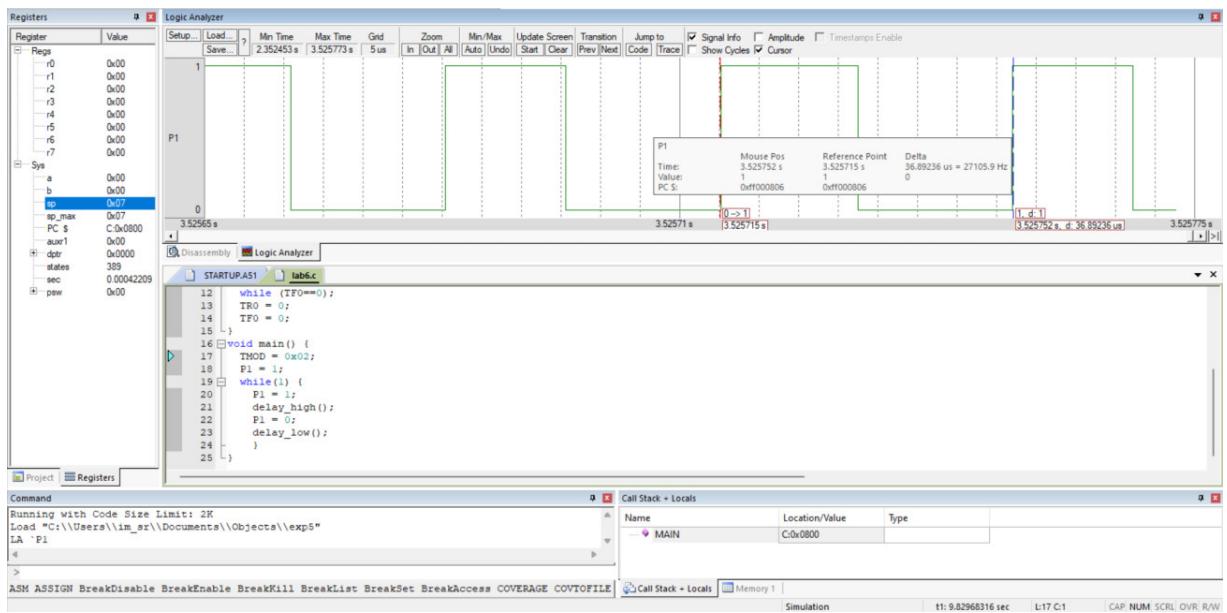


12) Using 8051, generate a square wave with a 75% duty cycle and 100KHz frequency. Use timer 0 in mode 2.

Code:

```
#include <reg51.h>
void delay_high() {
    TH0 = -7;
    TR0 = 1;
    while (TF0==0);
    TR0 = 0;
    TF0 = 0;
}
void delay_low() {
    TH0 = -2;
    TR0 = 1;
    while (TF0==0);
    TR0 = 0;
    TF0 = 0;
}
void main() {
    TMOD = 0x02;
    P1 = 1;
    while(1) {
        P1 = 1;
        delay_high();
        P1 = 0;
        delay_low();
    }
}
```

Output:



- 13) What value do we need to load the timer's register if we want to have a time delay of 5 ms?  
Show the program for timer 0 to create a pulse width of 5 ms on P2.3.

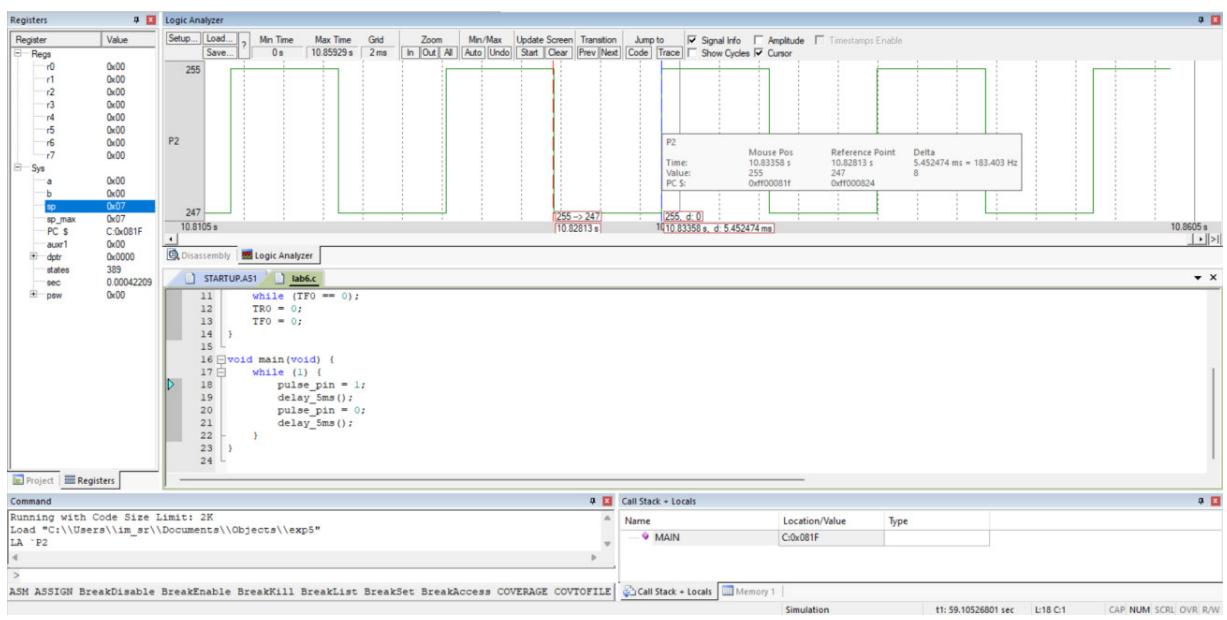
Code:

```
#include <reg51.h>
sbit pulse_pin = P2^3;
void delay_5ms(void) {
    TMOD = 0x01;
    TH0 = 0xEC;
    TL0 = 0x78;
    TR0 = 1;

    while (TF0 == 0);
    TR0 = 0;
    TF0 = 0;
}

void main(void) {
    while (1) {
        pulse_pin = 1;
        delay_5ms();
        pulse_pin = 0;
        delay_5ms();
    }
}
```

Output:



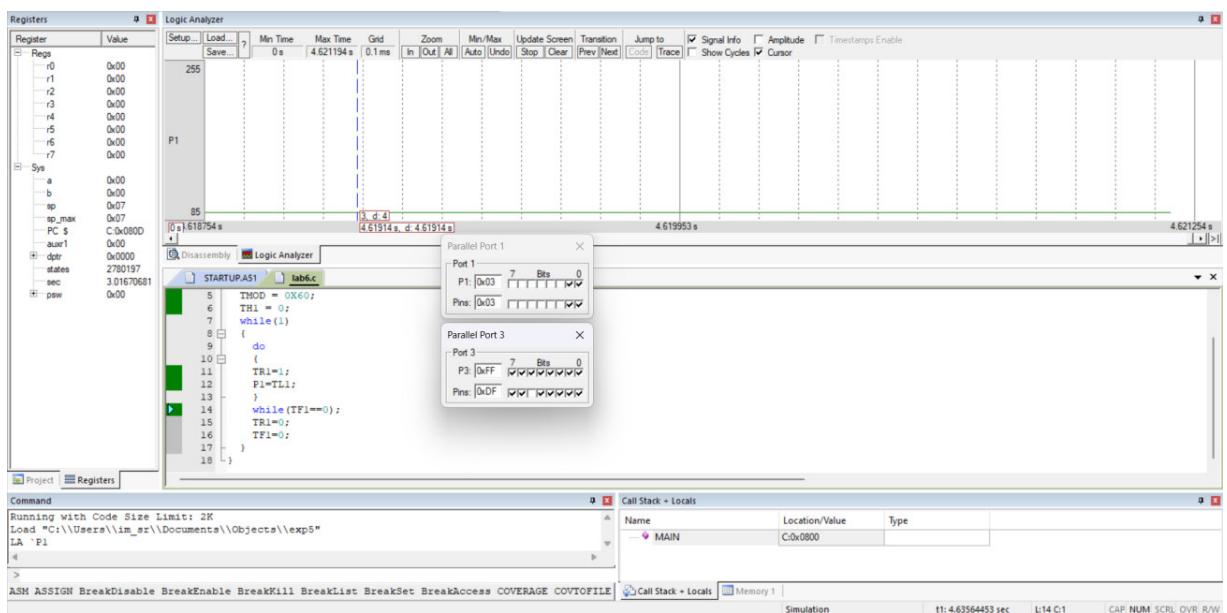
### Counters

- 14) Assume that a 1-Hz external clock is being fed into pin T1(P3.5). Write a C program for counter 1 in mode 2 to count up and display the state of the TL1 count on P1. Start the count at 0H.

Code:

```
#include <reg51.h>
void main(void)
{
    T1=1;
    TMOD = 0X60;
    TH1 = 0;
    while(1)
    {
        do
        {
            TR1=1;
            P1=TL1;
        }
        while(TF1==0);
        TR1=0;
        TF1=0;
    }
}
```

Output:

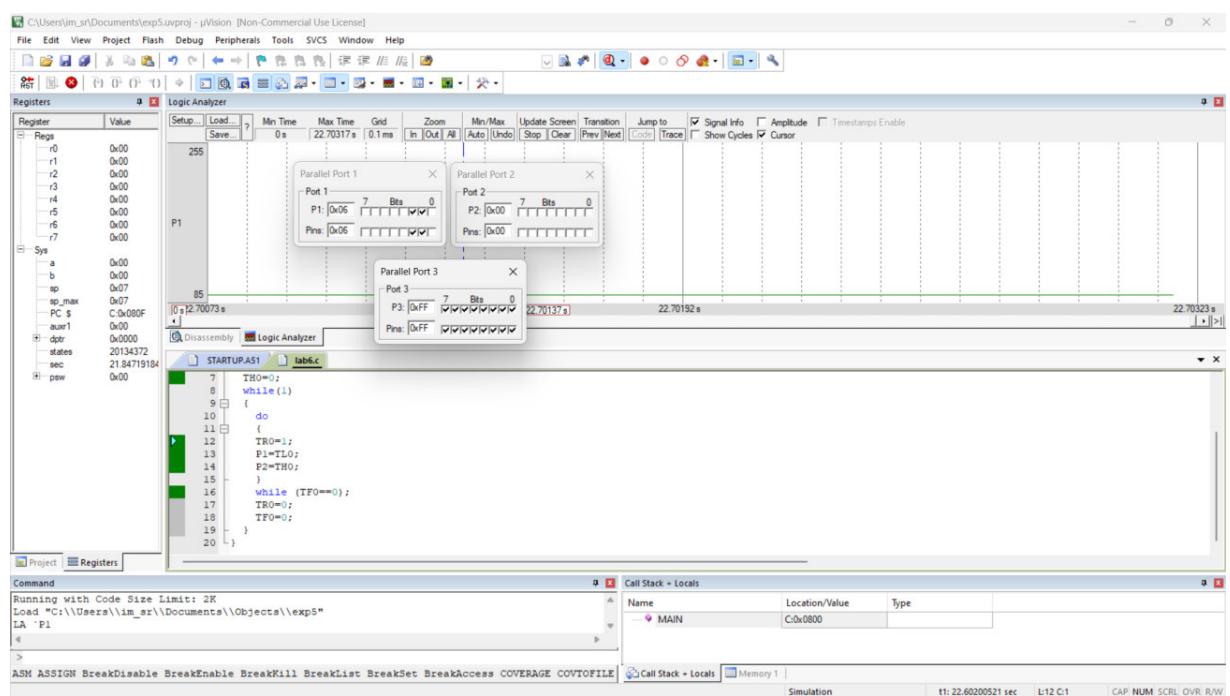


- 15) Assume that a 1-Hz external clock is being fed into pin T0(P3.4). Write a C program for counter 0 in mode 1 to count up and display the state of the TLO and TH0 registers on P2 and P1 respectively.

Code:

```
#include <reg51.h>
void main(void)
{
    T0=1;
    TMOD=0x05;
    TLO=0;
    TH0=0;
    while(1)
    {
        do
        {
            TR0=1;
            P1=TLO;
            P2=TH0;
        }
        while (TF0==0);
        TR0=0;
        TF0=0;
    }
}
```

Output:



## Exercise 7: Serial Communication Using 8051

Register Number: 21BEC1851

Date: 26-08-2024

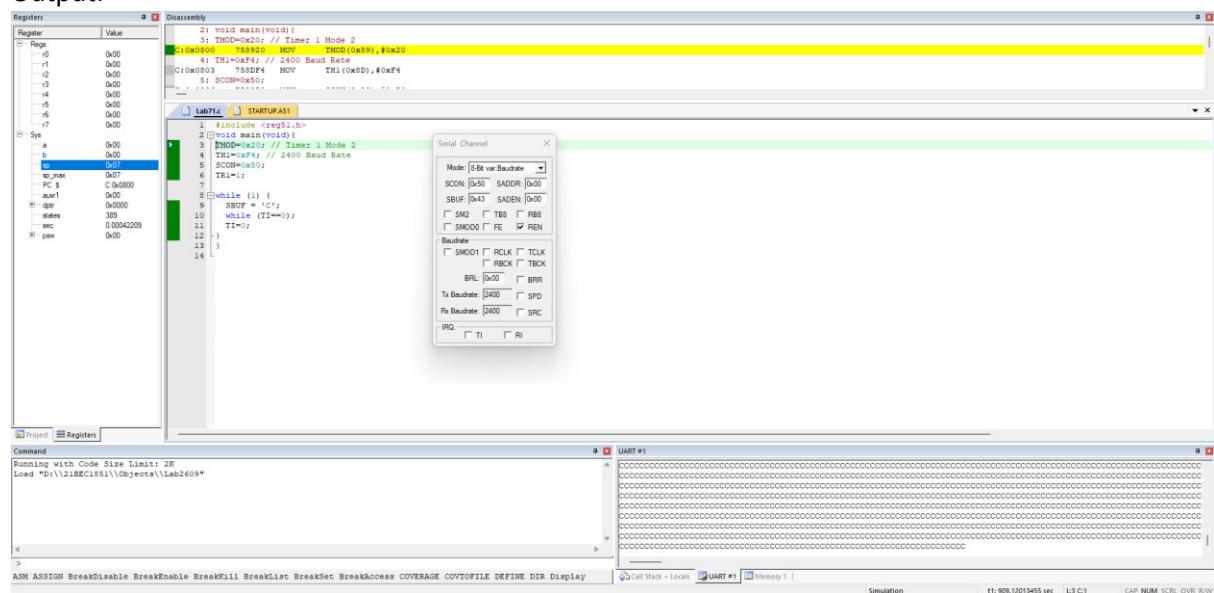
1. Write an 8051 C program to transfer serially the letter 'C' continuously at a 2400 baud rate.

Code:

```
#include <reg51.h>
void main(void){
    TMOD=0x20; // Timer 1 Mode 2
    TH1=0xF4; // 2400 Baud Rate
    SCON=0x50;
    TR1=1;

    while (1) {
        SBUF = 'C';
        while (TI==0);
        TI=0;
    }
}
```

Output:



2. Assume a switch is connected to pin P1.7. Write a program to monitor its status and send two messages to the serial port continuously as follows.

SW=0, send 'Yes'

SW = 1, send 'No'

Code:

```
#include <reg51.h>
void SerTx(unsigned char);
sbit pin1 = P1^7;
void main(void){
    TMOD=0x20;
    TH1=0xFD;
```

```

SCON=0x50;
TR1=1;
while (1) {
    if (pin1 == 0){
        SerTx('Y');
        SerTx('E');
        SerTx('S');
    }
    if (pin1 == 1){
        SerTx('N');
        SerTx('O');
    }
}
void SerTx(unsigned char x){
    SBUF=x;
    while (TI==0);
    TI=0;
}

```

### Output:

The screenshot shows a debugger interface with several windows:

- Registers:** Shows CPU registers (R0-R7, SP, PC, etc.) with their current values.
- Disassembly:** Shows the assembly code of the program. The highlighted line is:
 

```
32: while (TI==0); //wait until transmitted
```
- Symbols:** Shows the symbol table with entries like main, SBUF, and SerTx.
- Memory Dump:** Shows a large block of memory starting at address 0x00000000, mostly filled with zeros.
- Command Bar:** Includes buttons for Command, All Stack, Locals, UART #1, and Memory 1.
- Status Bar:** Displays Simulation, T1: 315.2332885 sec, L22 C1, CAP\_NUM SCR, D/R/W.

The screenshot shows a debugger interface with several windows:

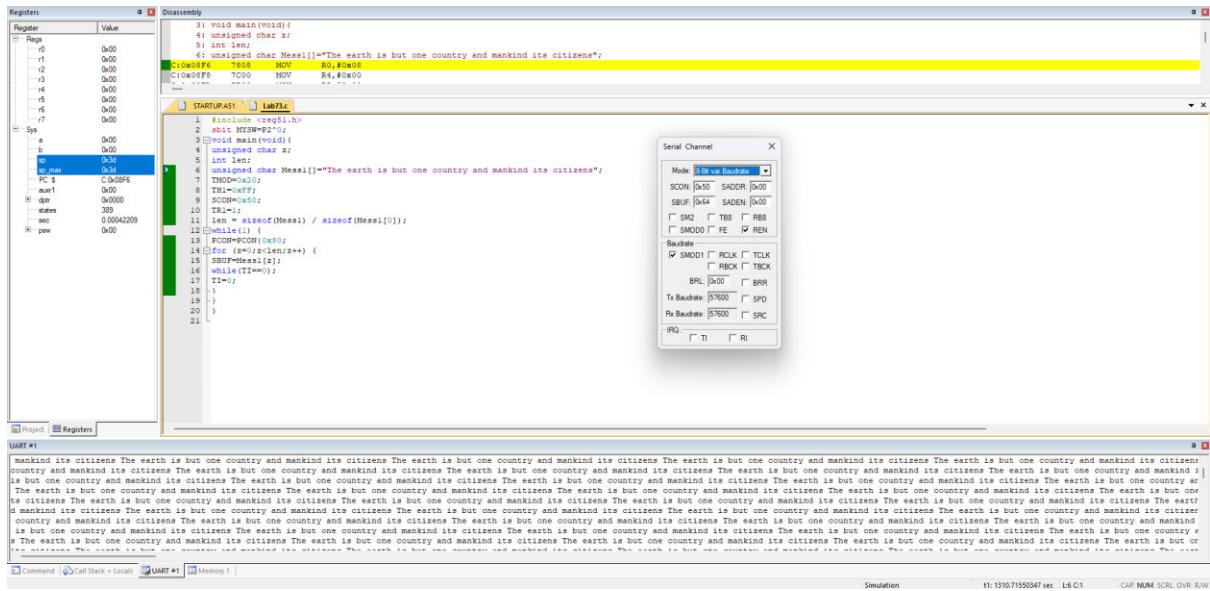
- Registers**: Shows CPU registers (r0-r7, r8-r15) all set to 0000.
- Memory**: Shows memory dump starting at address 0x00000000, filled with the byte 0x41 (ASCII 'A').
- Registers**: Shows CPU registers (r0-r7, r8-r15) all set to 0000.
- Disassembly**: Shows assembly code for a C program. The code initializes Timer 0, sets up serial port 1, and enters a loop where it repeatedly sends the character 'A' via serial port 1 until TI=0. The assembly code includes instructions like LDI, OUT, and SBI.
- Sys**: Shows system variables: a=0000, b=0000, m=0009, mc\_max=0009, mc\_s=0003A, auer1=0000, dptr=0000, sec=9603561409, pw=0000.
- Parallel Port 1**: Shows the state of parallel port pins P1[0:7] as 00000000.
- Port [0:7]**: Shows the state of port pins P0[0:7] as 11111111.

3. Write an 8051 C program to transfer serially the message “The earth is but one country and mankind its citizens” continuously at a 57,600 baud rate (PCON = PCON | 0x80;).

Code:

```
#include <reg51.h>
sbit MYSW=P2^0;
void main(void){
    unsigned char z;
    int len;
    unsigned char Mess1[]={The earth is but one country and mankind its citizens"};
    TMOD=0x20;
    TH1=0xFF;
    SCON=0x50;
    TR1=1;
    len = sizeof(Mess1) / sizeof(Mess1[0]);
    while(1) {
        PCON=PCON|0x80;
        for (z=0;z<len;z++) {
            SBUF=Mess1[z];
            while(TI==0);
            TI=0;
        }
    }
}
```

## Output:



- Write a program to send a message “We stand united” to the serial port. Assume a switch is connected to pin P1.3. Monitor its status and set the baud rate as follows.
- SW = 0, 9600 baud rate  
 SW = 1, 4800 baud rate

Code:

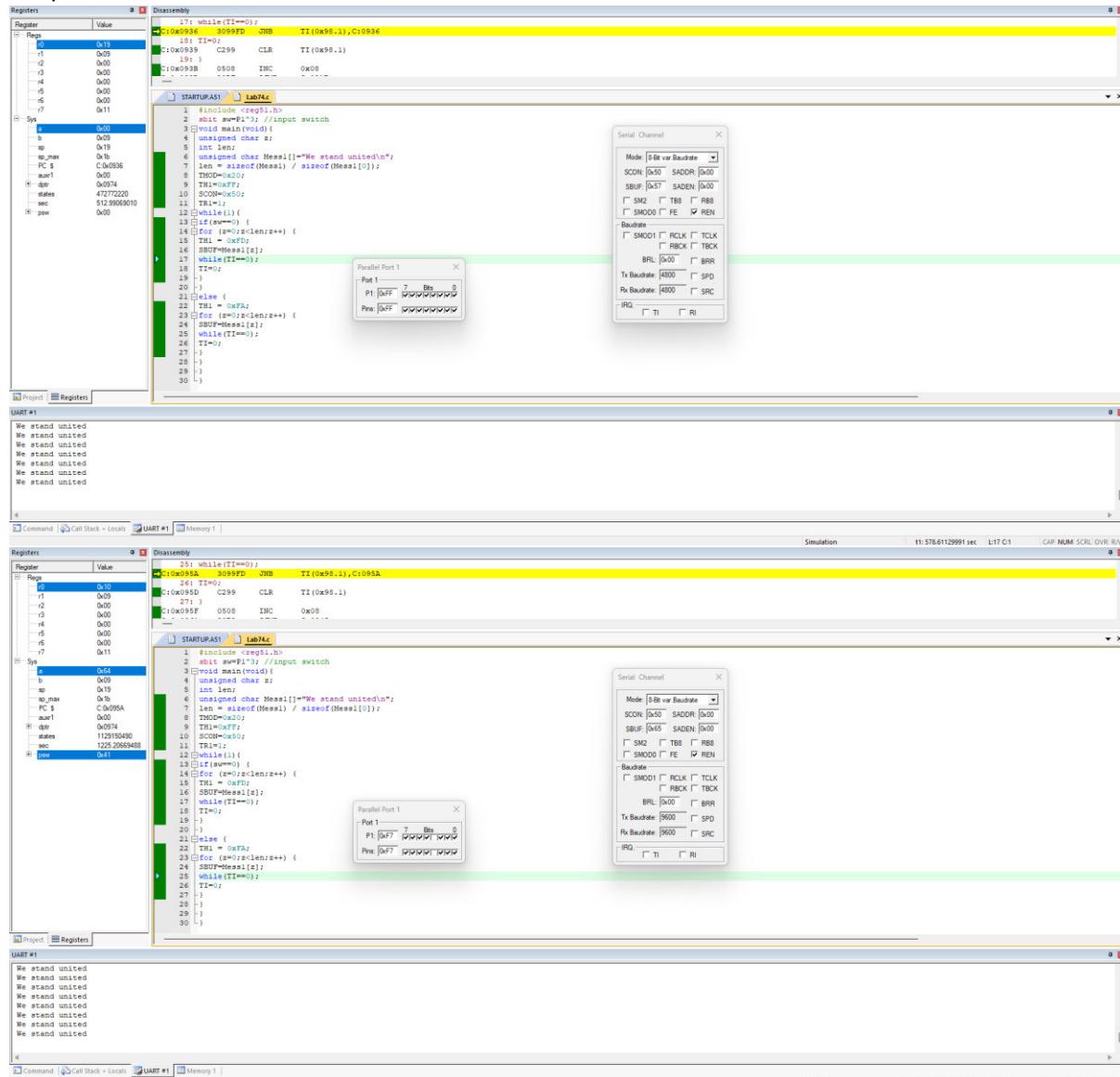
```

#include <reg51.h>
sbit sw=P1^3; //input switch
void main(void){
    unsigned char z;
    int len;
    unsigned char Mess1[]="We stand united\n";
    len = sizeof(Mess1) / sizeof(Mess1[0]);
    TMOD=0x20;
    TH1=0xFF;
    SCON=0x50;
    TR1=1;
    while(1){
        if(sw==0) {
            for (z=0;z<len;z++) {
                TH1 = 0xFD;
                SBUF=Mess1[z];
                while(TI==0);
                TI=0;
            }
        }
        else {
            TH1 = 0xFA;
            for (z=0;z<len;z++) {
                SBUF=Mess1[z];
                while(TI==0);
                TI=0;
            }
        }
    }
}

```

}

## Output:



5. Write an 8051 C program to receive bytes of data serially and put them in P1. Set the baud rate at 14 400, 8-bit data and 1 stop bit.

Code:

```
#include <reg51.h>
```

```
void main(void){
```

```
unsigned char mybyte;
```

TMOD=0x20:

TH1=0xFF;

SCON=0

TR1=1;

```
while (1) {
```

```
while (RI==0);
```

mybyte=SBUF;

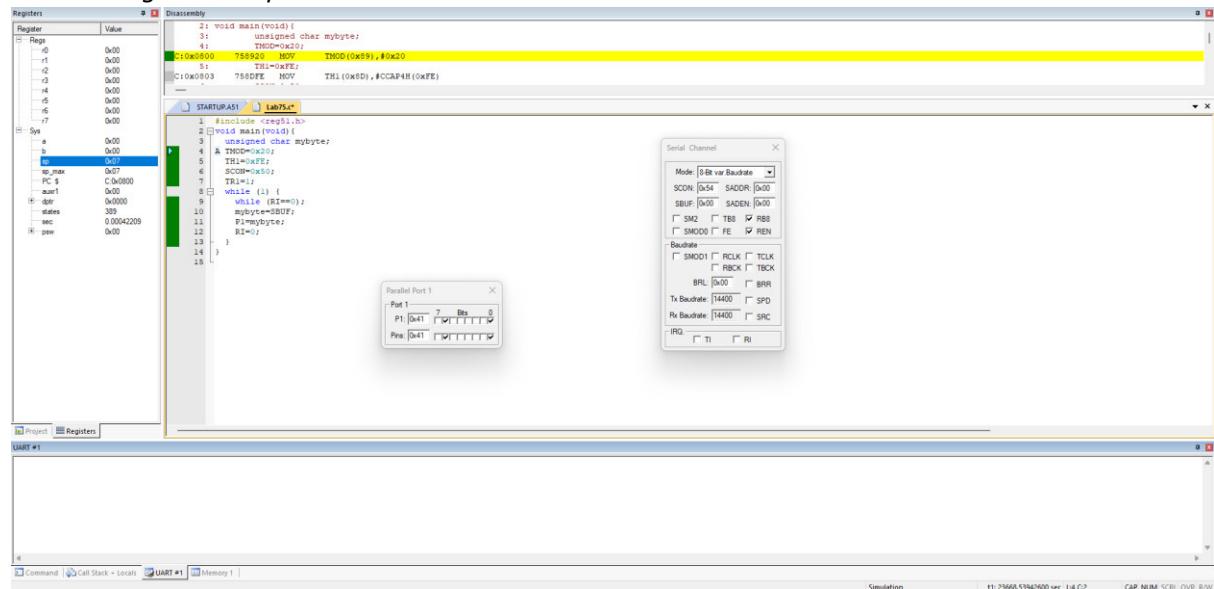
```

P1=mybyte;
RI=0;
}
}

```

Output:

Note: 'A' is given as input.



**Exercise 8:** Interrupts in 8051

**Register Number:** 21BEC1851

**Date:** 03-10-2024

1. Write an 8051 C program using interrupts to get data from P1 and send it to P2 while timer 0 generates a square wave of 3 kHz at P0.3. Use mode 1.

Code:

```
#include <reg51.h>

sbit square_wave = P0^3;

void timer0_ISR() interrupt 1 {

    square_wave = ~square_wave;

    TH0 = 0xFF; // Load high byte

    TL0 = 0x66; // Load low byte

}

void main() {

    P1 = 0xFF; // Set P1 as input

    P2 = 0x00; // Set P2 as output

    square_wave = 0; // Initialize P0.3 to low

    // Configure Timer 0 for mode 1 (16-bit timer)

    TMOD |= 0x01; // Timer 0 in mode 1

    TH0 = 0xFF; // Initial high byte value for 3 kHz square wave

    TL0 = 0x66; // Initial low byte value

    ET0 = 1; // Enable Timer 0 interrupt

    EA = 1; // Enable global interrupt

    TR0 = 1; // Start Timer 0

    while (1) {

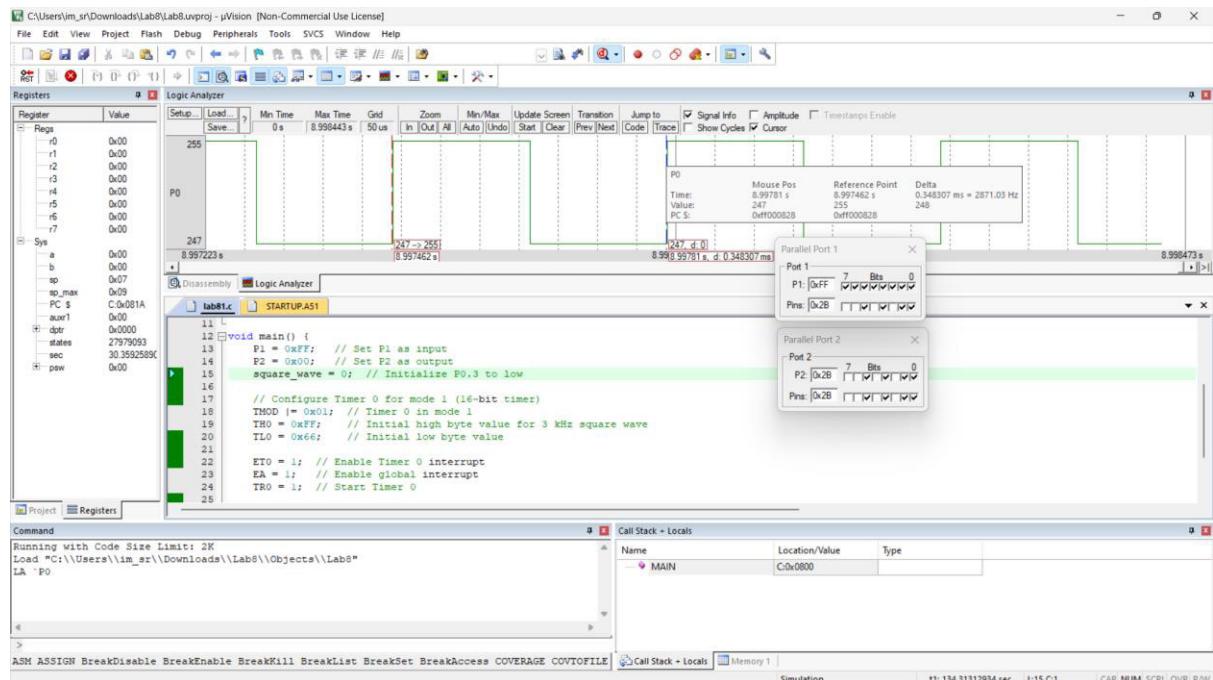
        P2 = P1;

    }

}
```

}

## Output:



2. Write an 8051 C program using interrupts to get data from P0 and send it to P1, while timer 1 turns on and off an LED connected to P2.3 every 50 ms.

Code:

```
#include <reg51.h>

sbit led = P2^3; // LED connected at P2.3

void timer1() interrupt 3 {
    led = ~led; // Toggle LED
    TH1 = 0x3C; // Reload for 50 ms delay
    TL1 = 0xB0;
}

void main() {
    TMOD = 0x10; // Timer 1 in Mode 1 (16-bit timer)
    IE = 0x88; // Enable Timer 1 interrupt
    TH1 = 0x3C; // Load high byte for 50 ms
    TL1 = 0xB0; // Load low byte for 50 ms
    TR1 = 1; // Start Timer 1

    while (1) {
```

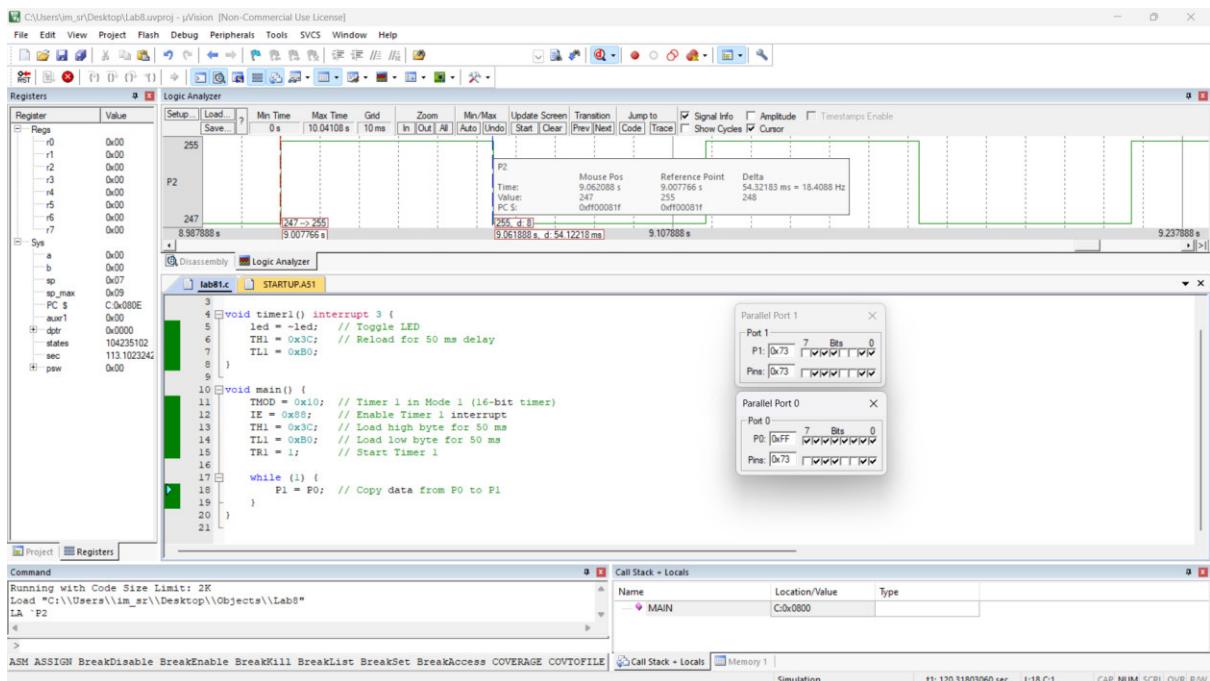
```

P1 = P0; // Copy data from P0 to P1
}

}

```

Output:



3. Write an 8051 C program using interrupts to toggle P1.7 every 20 ms (use timer 0 in mode 1) and to toggle P1.1 every 0.1 ms (use timer 1 in mode 2) simultaneously.

Code:

```

#include <reg51.h>

sbit toggle_20ms = P1^7; // P1.7 for toggling every 20 ms

sbit toggle_0_1ms = P1^1; // P1.1 for toggling every 0.1 ms

void timer0_ISR() interrupt 1 {

    toggle_20ms = ~toggle_20ms;

    TH0 = 0xB7; // Load high byte

    TL0 = 0xFF; // Load low byte
}

void timer1_ISR() interrupt 3 {

    // Toggle P1.1

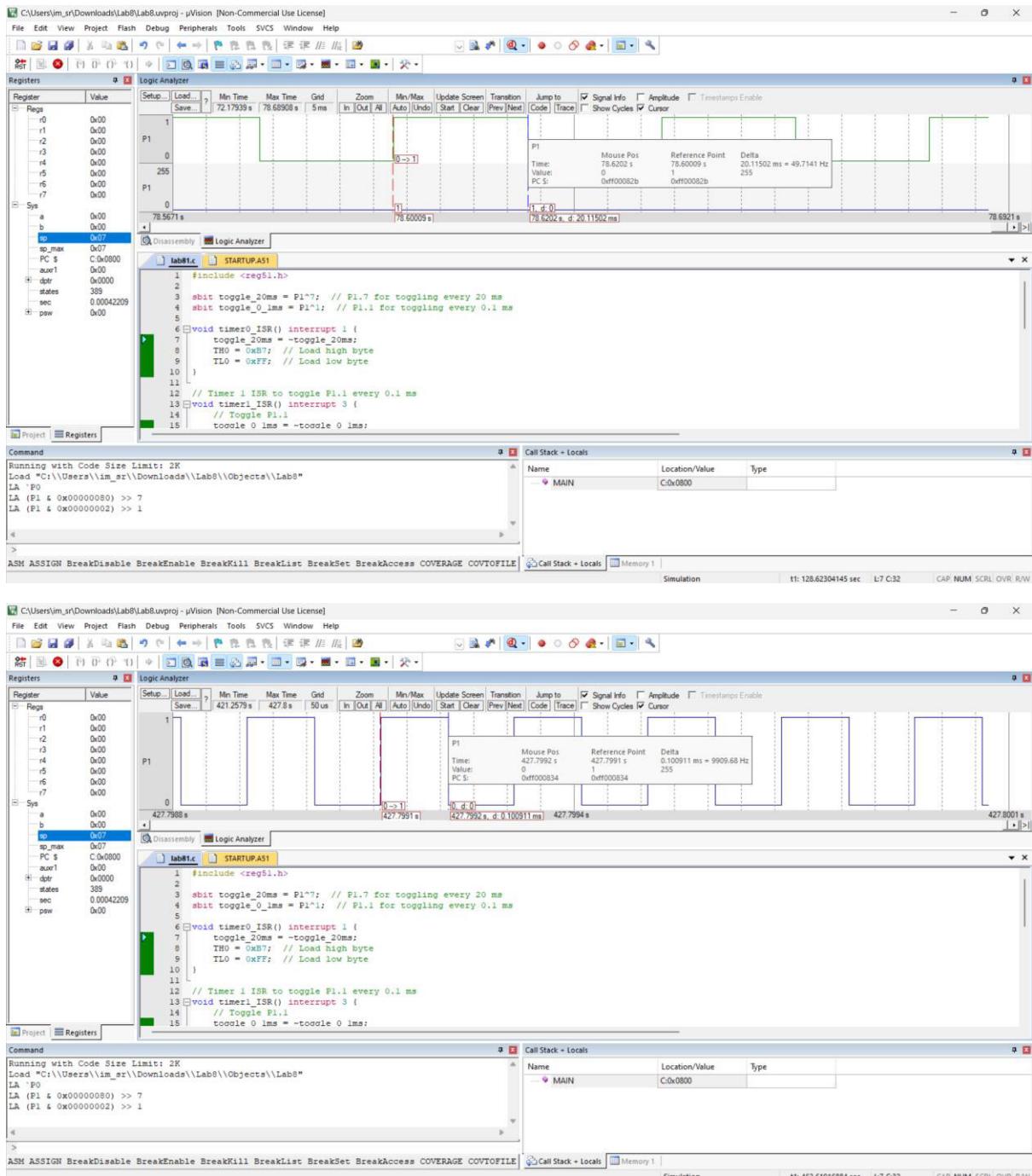
    toggle_0_1ms = ~toggle_0_1ms;

    TH1 = 0xA4; // Auto-reload value for 0.1 ms
}

```

```
void main() {  
    toggle_20ms = 0; // Set P1.7 to low  
    toggle_0_1ms = 0; // Set P1.1 to low  
  
    TMOD |= 0x01; // Set Timer 0 to mode 1  
    TH0 = 0xB7; // Initial high byte for 20 ms  
    TL0 = 0xFF; // Initial low byte  
  
    TMOD |= 0x20; // Set Timer 1 to mode 2  
    TH1 = 0xA4; // Auto-reload value for 0.1 ms  
  
    ET0 = 1; // Enable Timer 0 interrupt  
    ET1 = 1; // Enable Timer 1 interrupt  
    EA = 1; // Enable global interrupts  
  
    TR0 = 1; // Start Timer 0  
    TR1 = 1; // Start Timer 1  
  
    while (1) {  
    }  
}
```

Output:



4. Write an 8051 C program using interrupts to generate a clock signal of frequency 100kHz (use timer 0 in mode 2) at P2.2, while simultaneously monitoring P1.3 and sending it to P2.5.

Code:

```
#include <reg51.h>

sbit clockSignal = P2^2; // Clock signal at P2.2

sbit monitorInput = P1^3; // Monitor input at P1.3

sbit sendOutput = P2^5; // Send output at P2.5
```

```

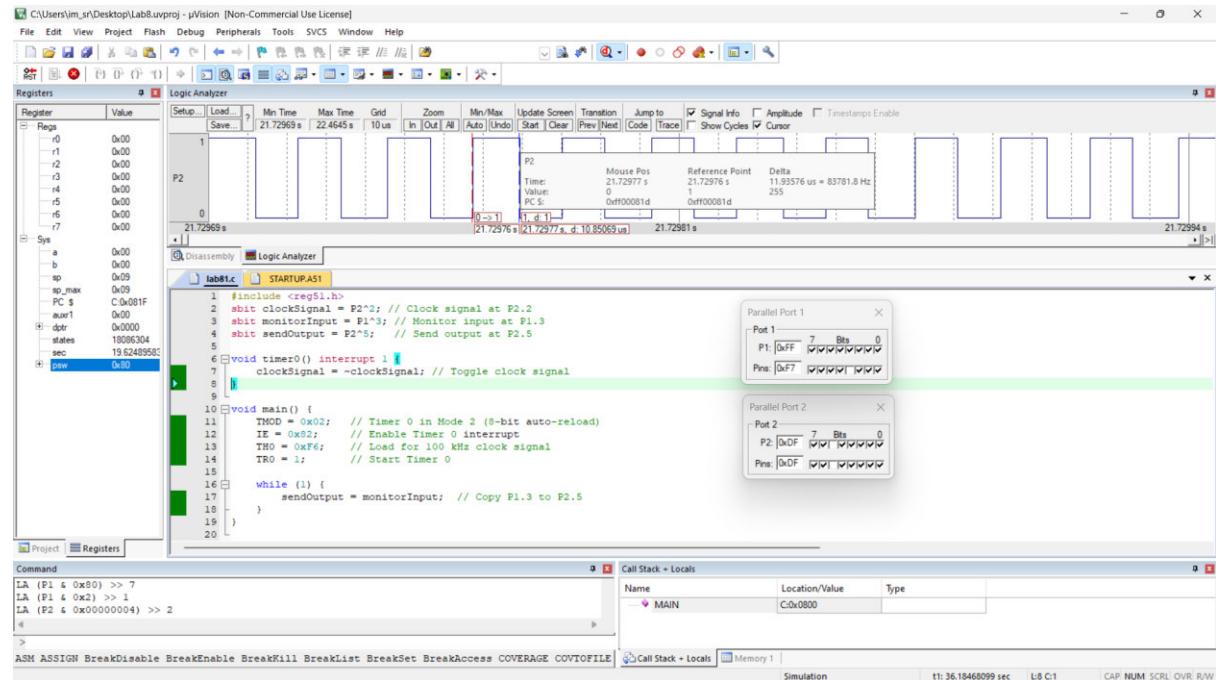
void timer0() interrupt 1 {
    clockSignal = ~clockSignal; // Toggle clock signal
}

void main() {
    TMOD = 0x02; // Timer 0 in Mode 2 (8-bit auto-reload)
    IE = 0x82; // Enable Timer 0 interrupt
    TH0 = 0xFC; // Load for 100 kHz clock signal
    TR0 = 1; // Start Timer 0

    while (1) {
        sendOutput = monitorInput; // Copy P1.3 to P2.5
    }
}

```

Output:



**Exercise 9:** Serial Communication Interrupts and External Interrupts

**Register Number:** 21BEC1851

**Date:** 10-10-2024

Serial Interrupts

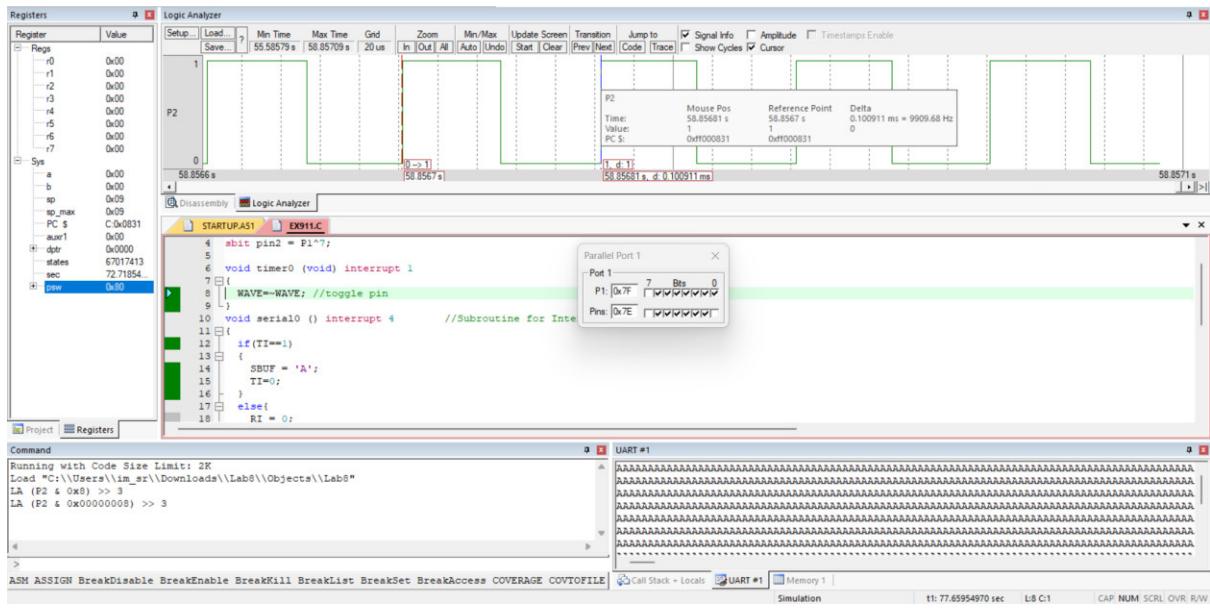
1. Write an 8051 C program to transfer a bit of data from P1.0 to P1.7, simultaneously generate a square wave of 100 $\mu$ s at pin P2.3, and send the letter 'A' to the serial port. Assume that XTAL = 11.0592 MHz. Use the 4800 baud rate.

Code:

```
#include <reg51.h>
sbit WAVE = P2^3;
sbit pin1 = P1^0;
sbit pin2 = P1^7;

void timer0 (void) interrupt 1
{
    WAVE=~WAVE; //toggle pin
}
void serial0 () interrupt 4      //Subroutine for Interrupt
{
    if(TI==1)
    {
        SBUF = 'A';
        TI=0;
    }
    else{
        RI = 0;
    }
}
void main()
{
    TH1=0xFA;          // Baud Rate: 4800
    TH0=0xD2;          // Timer 0 Auto reload
    TMOD=0x22;         // Setting mode 2 for timer 0 and timer 1
    SCON=0x50;
    TR0=1;
    TR1=1;
    IE=0x92;          // Enable Interrupt for serial port and timer 0
    while(1){
        pin2 = pin1;
    }
}
```

Output:



2. Write an 8051 C program to receive serially and send it to P0. Read port 1, serially transmit it and give a copy to P2. Assume that XTAL = 11.0592 MHz. Use the 2400 baud rate.

Code:

```
#include <reg51.h>
sbit WAVE =P0^1;
void ser_intr(void) interrupt 4      //Subroutine for Interrupt
{
if(TI==1)
{
TI=0;
}
else
{
char c;
c=SBUF;
P0=SBUF;

while(RI==0);
RI=0;
}
}

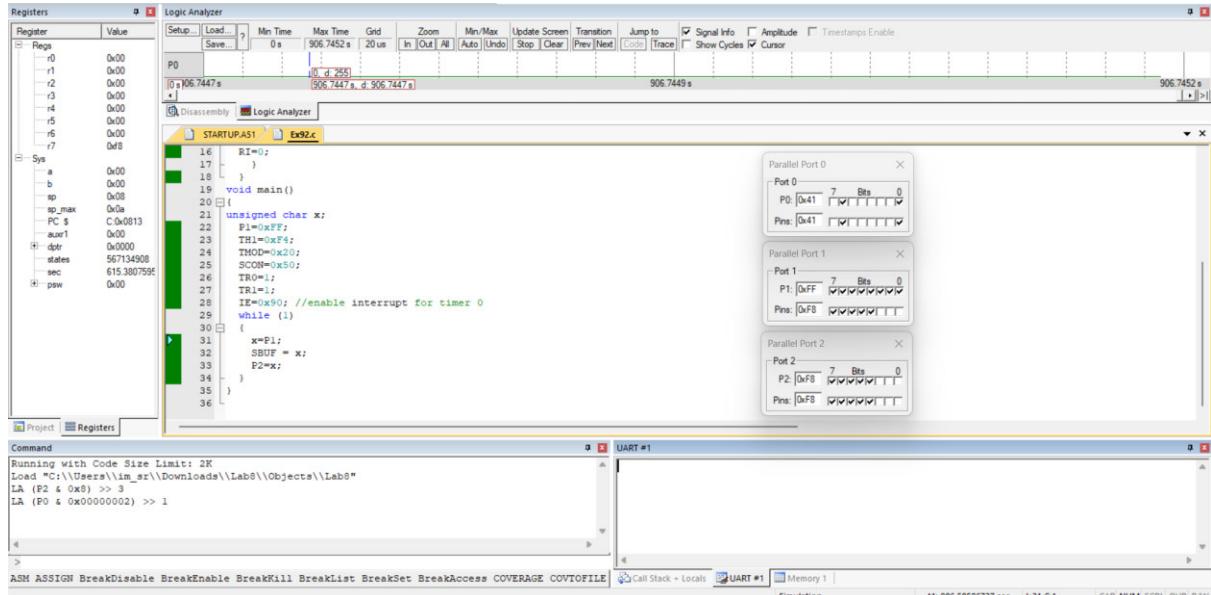
void main()
{
unsigned char x;
P1=0xFF;
TH1=0xF4;
TMOD=0x20;
SCON=0x50;
TR0=1;
TR1=1;
IE=0x90; //enable interrupt for timer 0
while (1)
```

```

{
    x=P1;
    SBUF = x;
    P2=x;
}

```

Output:



### External Interrupts

3. Write an 8051 C program using interrupts to generate 10kHz on P2.1 using T0 (8-bit auto-reload) and to count the pulses at EXT1 (edge-triggered) and display them on P0.

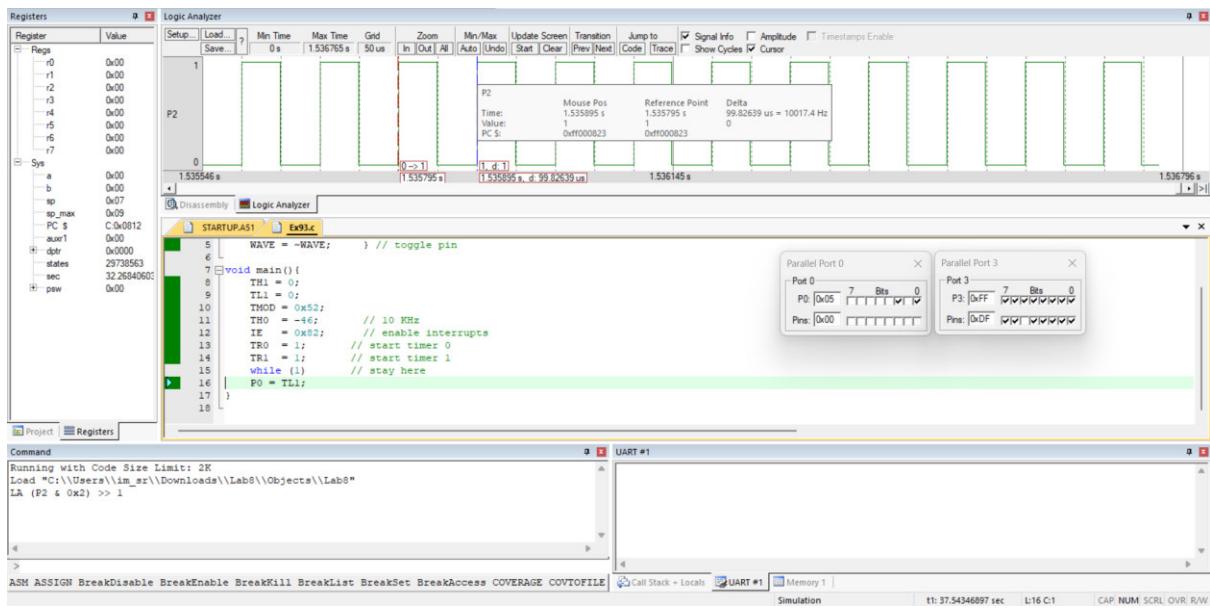
Code:

```

#include <reg51.h>
sbit WAVE = P2^1;
void timer0() interrupt 1
{
    WAVE = ~WAVE;           // toggle pin
}
void main(){
    TH1 = 0;
    TL1 = 0;
    TMOD = 0x52;
    TH0 = -46;             // 10 KHz
    IE = 0x82;              // enable interrupts
    TR0 = 1;                // start timer 0
    TR1 = 1;                // start timer 1
    while (1)
        // stay here
    P0 = TL1;
}

```

Output:



4. Write an 8051 C program using interrupts to count the pulses at EXTO in level-triggered mode and display them on P1. Simultaneously, serially transmit the letter 'V' at a 4800 baud rate.

Code:

```
#include <reg51.h>
```

```
unsigned int ext0_count = 0;
```

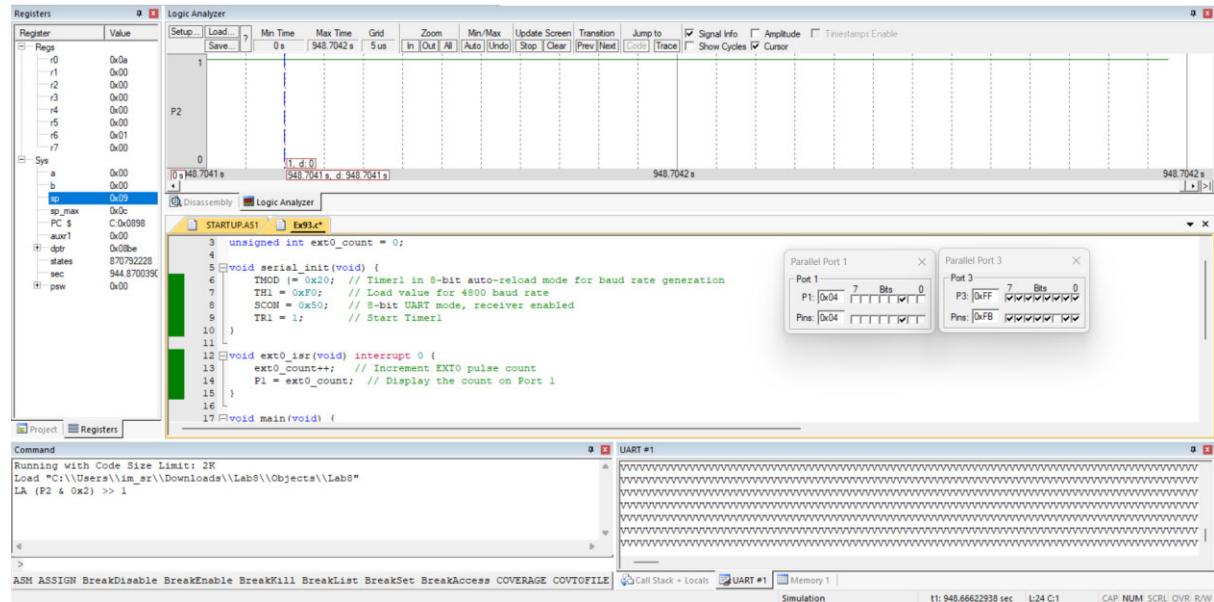
```
void serial_init(void) {
    TMOD |= 0x20; // Timer1 in 8-bit auto-reload mode for baud rate generation
    TH1 = 0xF0; // Load value for 4800 baud rate
    SCON = 0x50; // 8-bit UART mode, receiver enabled
    TR1 = 1; // Start Timer1
}
```

```
void ext0_isr(void) interrupt 0 {
    ext0_count++; // Increment EXTO pulse count
    P1 = ext0_count; // Display the count on Port 1
}
```

```
void main(void) {
    P1 = 0x00; // Initialize Port 1
    IE = 0x81; // Enable external interrupt 0 (level-triggered mode)
    serial_init(); // Initialize serial communication

    while (1) {
        SBUF = 'V'; // Send 'V' over serial
        while (!TI); // Wait for transmission to complete
        TI = 0; // Clear transmit interrupt flag
    }
}
```

## Output:



**Exercise 10:** External Hardware Interrupts, LCD and Keypad

**Register Number:** 21BEC1851

**Date:** 24-10-2024

External Hardware Interrupt

1. Write an 8051 C program to control an LED using external interrupt INT0. Simultaneously make another LED toggle with some delay.

Code:

```
#include <reg51.h>
sbit SW =P3^2;
sbit LED1=P1^5;
sbit LED2=P1^1;
void extint0() interrupt 0
{
    LED2=0;
}
void main()
{
    unsigned int i;
    SW=1;
    IE=0x81;
    while(1)
    {
        LED1=0;
        for(i=10000;i>0;i--);
        LED1=1;
        for(i=10000;i>0;i--);
    }
}
```

Keypad

2. Write an 8051 C program to interface a 4x4 keypad and display the pressed key on an LCD.

Code:

```
#include<reg51.h>
#define display_port P2 //Data pins connected to port 2 on microcontroller
sbit rs = P3^7; //RS pin connected to pin 2 of port 3
sbit rw = P3^6; // RW pin connected to pin 3 of port 3
sbit e = P3^5; //E pin connected to pin 4 of port 3
sbit C4 = P0^3; // Connecting keypad to Port 1
sbit C3 = P0^2;
sbit C2 = P0^1;
sbit C1 = P0^0;
sbit R4 = P1^3;
sbit R3 = P1^2;
sbit R2 = P1^1;
sbit R1 = P1^0;

void msdelay(unsigned int time) // Function for creating delay in milliseconds.
{
```

```

unsigned i,j ;
for(i=0;i<time;i++)
for(j=0;j<1275;j++);
}

void lcd_cmd(unsigned char command) //Function to send command instruction to LCD
{
    display_port = command;
    rs= 0;
    rw=0;
    e=1;
    msdelay(1);
    e=0;
}

void lcd_data(unsigned char disp_data) //Function to send display data to LCD
{
    display_port = disp_data;
    rs= 1;
    rw=0;
    e=1;
    msdelay(1);
    e=0;
}

void lcdstring(char *str)
{
    while(*str)
    {
        lcd_data(*str);
        str++;
    }
}

void lcd_init() //Function to prepare the LCD and get it ready
{
    lcd_cmd(0x38); // for using 2 lines and 5X7 matrix of LCD
    msdelay(10);
    lcd_cmd(0x0F); // turn display ON, cursor blinking
    msdelay(10);
    lcd_cmd(0x01); //clear screen
    msdelay(10);
    lcd_cmd(0x80); // bring cursor to position 1 of line 1
    msdelay(10);
}

void row_finder1() //Function for finding the row for column 1
{

```

```
R1=R2=R3=R4=1;  
C1=C2=C3=C4=0;  
if(R1==0)  
lcd_data('0');  
if(R2==0)  
lcd_data('4');  
if(R3==0)  
lcd_data('8');  
if(R4==0)  
lcd_data('C');  
}
```

```
void row_finder2() //Function for finding the row for column 2  
{  
R1=R2=R3=R4=1;  
C1=C2=C3=C4=0;  
if(R1==0)  
lcd_data('1');  
if(R2==0)  
lcd_data('5');  
if(R3==0)  
lcd_data('9');  
if(R4==0)  
lcd_data('D');  
}
```

```
void row_finder3() //Function for finding the row for column 3  
{  
R1=R2=R3=R4=1;  
C1=C2=C3=C4=0;  
  
if(R1==0)  
lcd_data('2');  
if(R2==0)  
lcd_data('6');  
if(R3==0)  
lcd_data('A');  
if(R4==0)  
lcd_data('E');  
}
```

```
void row_finder4() //Function for finding the row for column 4  
{  
R1=R2=R3=R4=1;  
C1=C2=C3=C4=0;  
if(R1==0)  
lcd_data('3');  
if(R2==0)
```

```

lcd_data('7');
if(R3==0)
lcd_data('B');
if(R4==0)
lcd_data('F');
}

void main()
{
lcd_init();
lcdstring("Please press key");
lcd_cmd(0xC0);

while(1)
{
msdelay(30);
C1=C2=C3=C4=1;
R1=R2=R3=R4=0;
if(C1==0)
row_finder1();
else if(C2==0)
row_finder2();
else if(C3==0)
row_finder3();
else if(C4==0)
row_finder4();
}
}

```

### LCD

3. Write an 8051 C program to interface an LCD and display the word "SENSE" on the screen.

Code:

```

#include <reg51.h>
void LCD_CMD(unsigned char CMD);
void LCD_DATA(unsigned char DATA);
void DELAY_ms(unsigned char j);

sbit RS= P3^7;
sbit RW= P3^6;
sbit EN= P3^5;

void main()
{
P2=0x00;           //Port used to connect LCD datapins D0-D7
LCD_CMD(0x01);    //Clear the display screen
DELAY_ms(5);
LCD_CMD(0x0E);    //Display on, cursor blinking
DELAY_ms(5);

```

```

LCD_CMD(0x38);           //2 lines, 5x8 matrix, 8bit mode
DELAY_ms(5);
LCD_CMD(0x80);           //Force the cursor to the beginning of the 1st line
DELAY_ms(5);
LCD_DATA('S');
LCD_DATA('E');
LCD_DATA('N');
LCD_DATA('S');
LCD_DATA('E');
while(1);
}

void LCD_CMD(unsigned char CMD)
{
    P2=CMD;
    RS=0;
    RW=0;
    EN=1;
    DELAY_ms(5);
    EN=0;
}

void LCD_DATA(unsigned char DATA)
{
    P2=DATA;
    RS=1;
    RW=0;
    EN=1;
    DELAY_ms(5);
    EN=0;
}

void DELAY_ms(unsigned int j)
{
    unsigned int i;
    for(j>0;j--)
    {
        for(i=250;i>0;i--);
        for(i=250;i>0;i--);
    }
}

```

4. Write an 8051 C Program to display "Welcome To VIT" in first row and "SENSE" in 2nd row of the LCD all characters at once (8-bit Mode)

Code:

```
#include <reg51.h>
void LCD_CMD(unsigned char CMD);
void LCD_DATA(unsigned char DATA);
```

```

void DELAY_ms(unsigned char j);

sbit RS= P3^7;
sbit RW= P3^6;
sbit EN= P3^5;

void main()
{
    unsigned char i,message1[]={ "Welcome to VIT"};
    unsigned char k,message2[]={ "SENSE"};
    P2=0x00;           //Port used to connect LCD datapins D0-D7
    LCD_CMD(0x01);      //Clear the display screen
    DELAY_ms(5);
    LCD_CMD(0x0E);      //Display on, cursor blinking
    DELAY_ms(5);
    LCD_CMD(0x38);      //2 lines, 5x8 matrix, 8bit mode
    DELAY_ms(5);
    LCD_CMD(0x80);      //Force the cursor to the beginning of the 1st line
    DELAY_ms(5);
    for(i=0;message1[i]!=0;i++)
    {
        LCD_DATA(message1[i]);
    }
    DELAY_ms(5);
    LCD_CMD(0xC5);      //Force the cursor to the beginning of the 2nd line
    DELAY_ms(5);
    for(k=0;message2[k]!=0;k++)
    {
        LCD_DATA(message2[k]);
    }
    while(1);
}

void LCD_CMD(unsigned char CMD)
{
    P2=CMD;
    RS=0;
    RW=0;
    EN=1;
    DELAY_ms(5);
    EN=0;
}

void LCD_DATA(unsigned char DATA)
{
    P2=DATA;
    RS=1;
    RW=0;
}

```

```
EN=1;
DELAY_ms(5);
EN=0;
}

void DELAY_ms(unsigned int j)
{
    unsigned int i;
    for(j>0;j--)
    {
        for(i=250;i>0;i--);
        for(i=250;i>0;i--);
    }
}
```

## Exercise 11: Waveform Generation and Stepper Motor

Register Number: 21BEC1851

Date: 07-11-2024

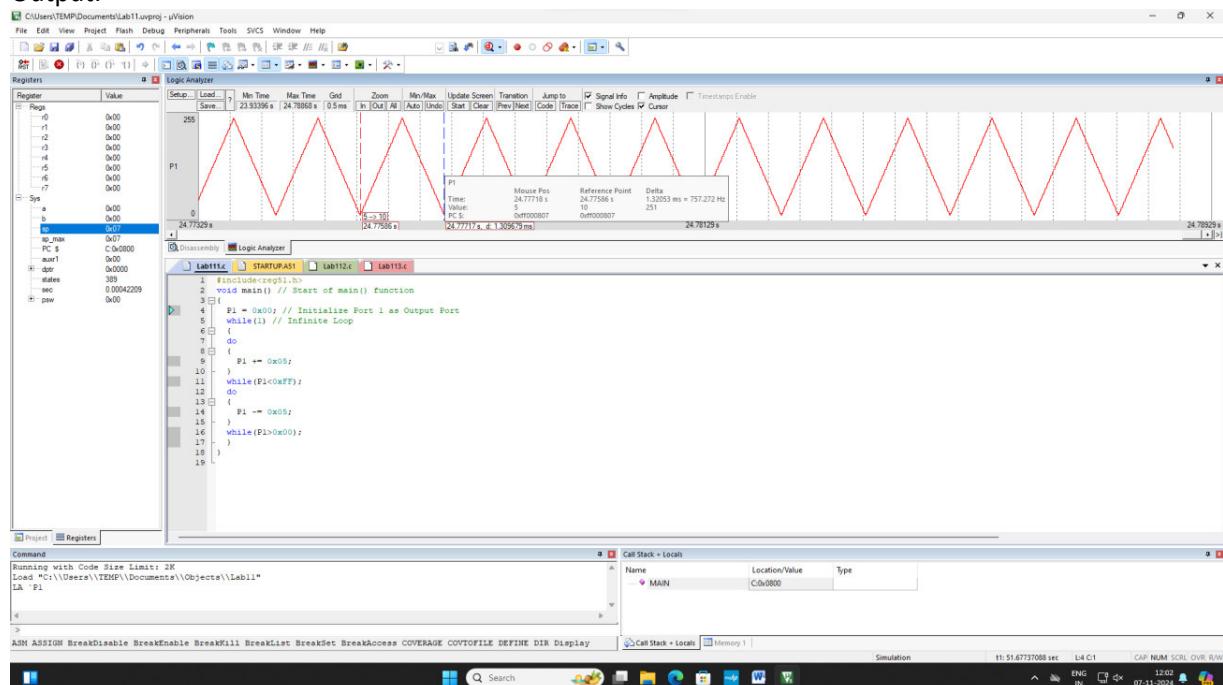
### Waveform Generation

#### 1. Triangular wave generation

Code:

```
#include<reg51.h>
void main() // Start of main() function
{
    P1 = 0x00; // Initialize Port 1 as Output Port
    while(1) // Infinite Loop
    {
        do
        {
            P1 += 0x05;
        }
        while(P1<0xFF);
        do
        {
            P1 -= 0x05;
        }
        while(P1>0x00);
    }
}
```

#### Output:

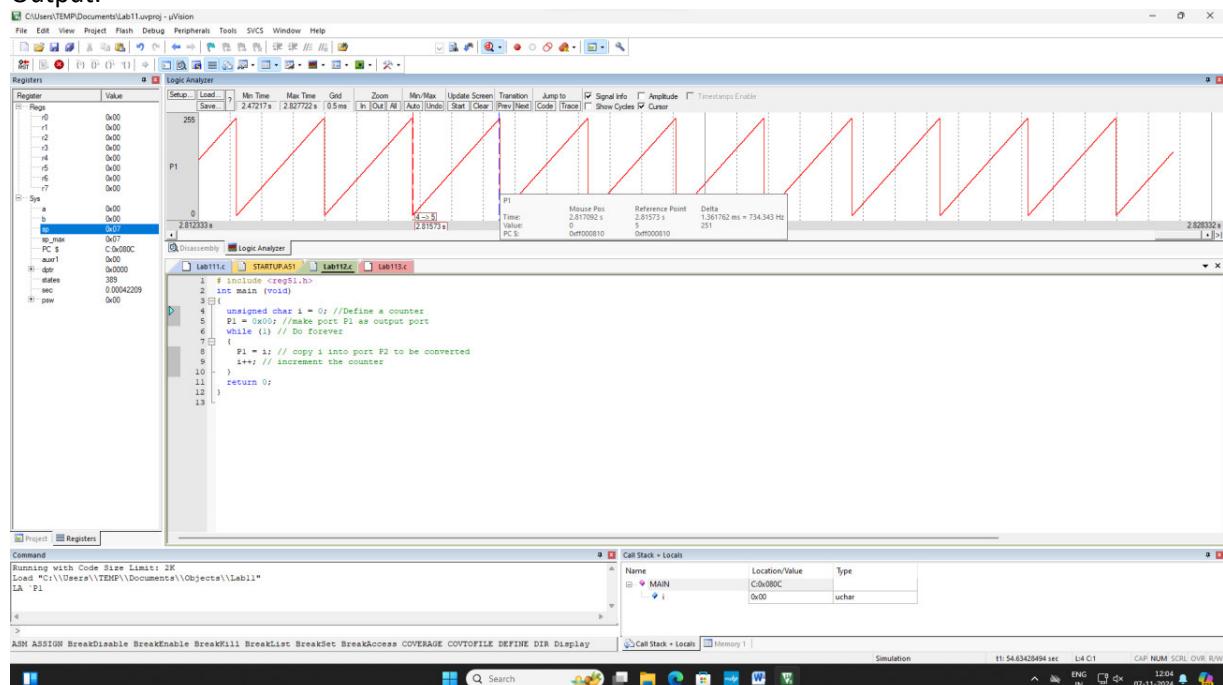


#### 2. Sawtooth wave generation

Code:

```
# include <reg51.h>
int main (void)
{
    unsigned char i = 0; //Define a counter
    P1 = 0x00; //make port P1 as output port
    while (1) // Do forever
    {
        P1 = i; // copy i into port P2 to be converted
        i++; // increment the counter
    }
    return 0;
}
```

Output:

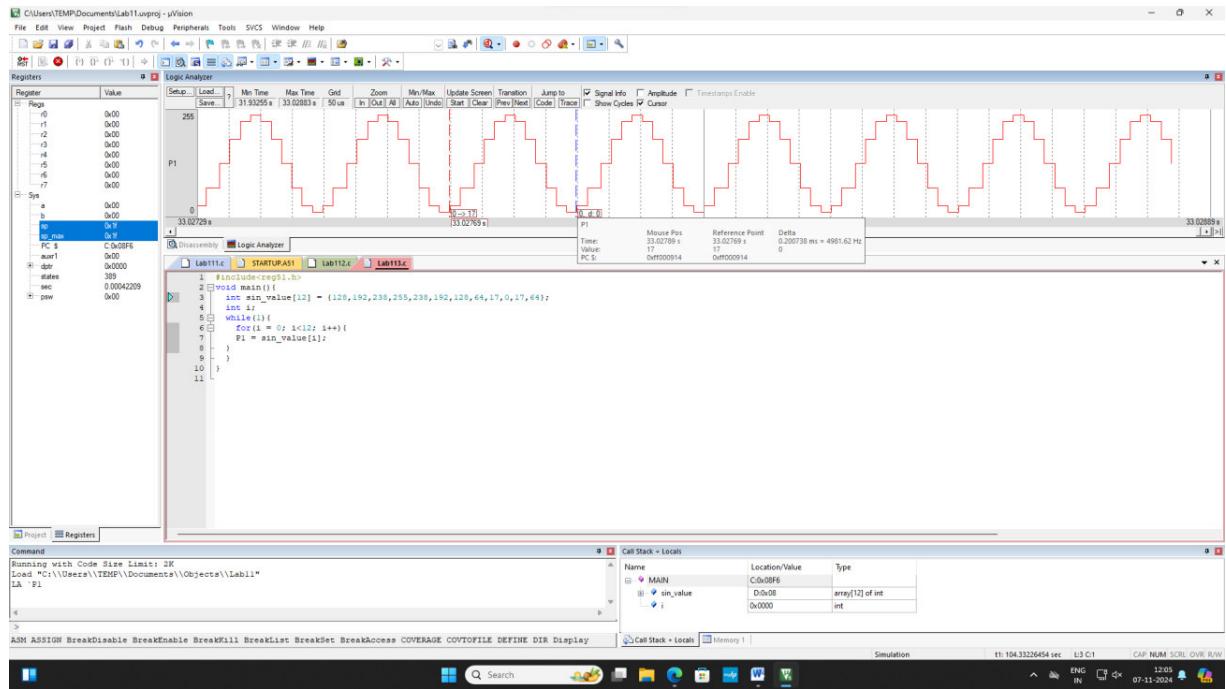


### 3. Sine wave generation

Code:

```
#include<reg51.h>
void main(){
    int sin_value[12] = {128,192,238,255,238,192,128,64,17,0,17,64};
    int i;
    while(1){
        for(i = 0; i<12; i++){
            P1 = sin_value[i];
        }
    }
}
```

Output:



# Stepper Motor

Code:

```
#include <reg51.h>
#define Stepper_Port P0 /* Define Stepper Motor Port */
/* Function to provide delay of 1ms at 11.0592 MHz */
void delay(unsigned int count)
{
    int i,j;
    for(i=0; i<count; i++)
        for(j=0; j<112; j++);
}
int main(void)
{
    int i,period;
    period = 100; /* Set period in between two steps of Stepper Motor */
    while (1)
    {
        /* Rotate Stepper Motor clockwise with Half step sequence */
        for(i=0; i<12; i++)
        {
            Stepper_Port = 0x09;
            delay(period);
            Stepper_Port = 0x08;
            delay(period);
            Stepper_Port = 0x0C;
            delay(period);
        }
    }
}
```

```
Stepper_Port = 0x04;
delay(period);
Stepper_Port = 0x06;
delay(period);
Stepper_Port = 0x02;
delay(period);
Stepper_Port = 0x03;
delay(period);
Stepper_Port = 0x01;
delay(period);
}
/* last one step to acquire initial position */
Stepper_Port = 0x09;
delay(period);
delay(1000);
/* Rotate Stepper Motor Anticlockwise with Full step sequence */
for(i=0; i<12; i++)
{
    Stepper_Port = 0x09;
    delay(period);
    Stepper_Port = 0x03;
    delay(period);
    Stepper_Port = 0x06;
    delay(period);
    Stepper_Port = 0x0C;
    delay(period);
}
Stepper_Port = 0x09;
delay(period);
delay(1000);
}
}
```