

**SWT 11022 PRACTICAL FOR FUNDAMENTALS OF
PROGRAMMING
DEPARTMENT OF ICT
FACULTY OF TECHNOLOGY
SOUTH-EASTERN UNIVERSITY OF SRI LANKA**

PREPARING AND SUBMITTING THE LAB RECORD – 05

Registration Number – SEU/IS/20/ICT/019

Academic Year – 2020/2021

Date – 2022/11/17

Practical Number – 05_Thursday

Title – Repetition (for loop)

Aim(s) –

- Dealing with for loop and nested for loop

TASKS

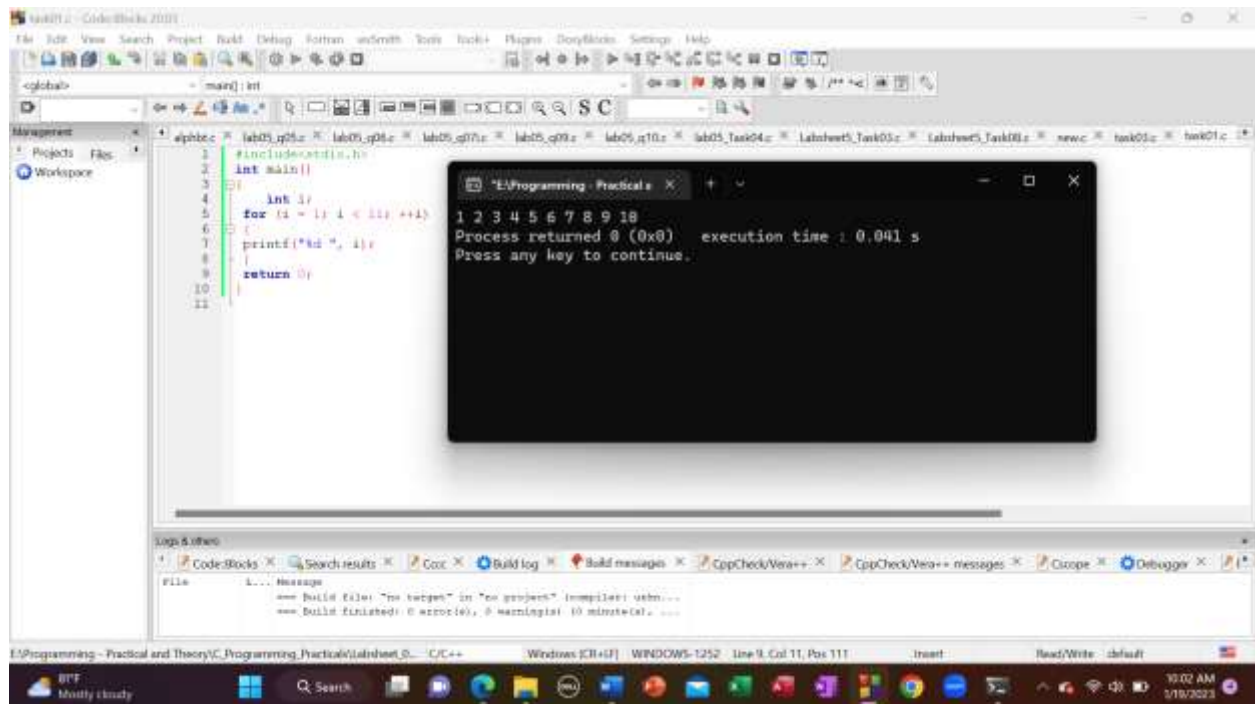
01.

```
#include<stdio.h>

int main()
{
    int i;

    for (i = 1; i < 11; ++i)
    {
        printf("%d ", i);
    }

    return 0;
}
```



The screenshot shows the Code::Blocks IDE with a C++ project. The source code in 'task01.c' is as follows:

```
1 #include<stdio.h>
2 int main()
3 {
4     int i;
5     for (i = 1; i < 11; ++i)
6     {
7         printf("%d ", i);
8     }
9     return 0;
10 }
11
```

The output window displays the following text:

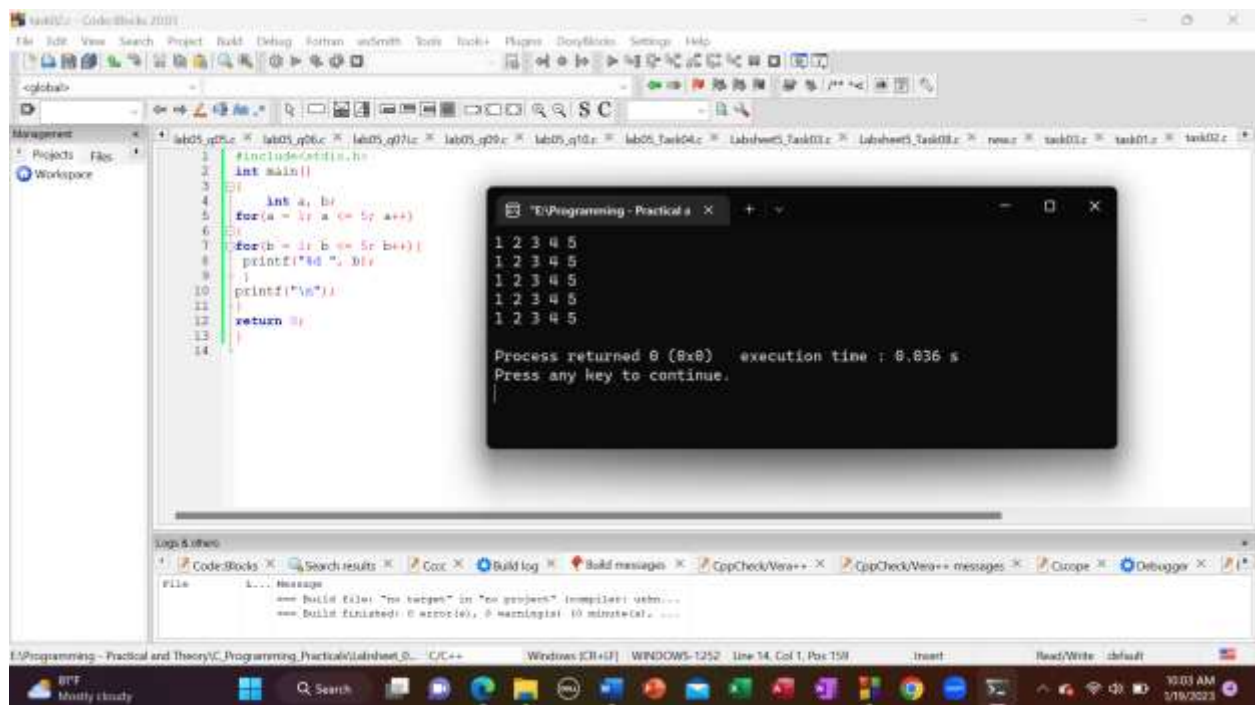
```
1 2 3 4 5 6 7 8 9 10
Process returned 0 (0x0)   execution time : 0.041 s
Press any key to continue.
```

The status bar at the bottom indicates the file path: 'E:\Programming - Practical and Theory\Programming_Practicals\Task01_S...'. The taskbar shows the system clock as 10:02 AM on 1/18/2023.

02.

```
#include<stdio.h>

int main()
{
    int a, b;
    for(a = 1; a <= 5; a++)
    {
        for(b = 1; b <= 5; b++){
            printf("%d ", b);
        }
        printf("\n");
    }
    return 0;
}
```



The screenshot displays the CodeBlocks IDE interface. The main editor window shows a C program that prints a 5x5 grid of numbers from 1 to 5. The program is as follows:

```
1 #include<stdio.h>
2 int main()
3 {
4     int a, b;
5     for(a = 1; a <= 5; a++)
6     {
7         for(b = 1; b <= 5; b++){
8             printf("%d ", b);
9         }
10        printf("\n");
11    }
12    return 0;
13 }
14
```

Below the code editor, a terminal window titled "T:\Programming - Practical 5" shows the output of the program:

```
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5

Process returned 0 (0x0)   execution time : 0.036 s
Press any key to continue.
```

The bottom status bar of the IDE indicates the file path: "I:\Programming - Practical and Theory\Programming Practical\labSheet_0... \C\C++". The Windows taskbar at the very bottom shows the date and time as 10:03 AM on 3/19/2023.

03.

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

```
int main()
```

```
{
```

```
    int i,sum=0;
```

```
    for (i=1;i<100;i+=2)
```

```
    {
```

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

```
        sum=sum+i;
```

```
    }
```

```
    printf("\nSum is : %d",sum);
```

```
    return 0;
```

```
}
```

The screenshot shows a C++ IDE with the following components:

- Source Editor:** Contains the C++ code for calculating the sum of odd numbers from 1 to 99. The code is as follows:

```
#include <stdio.h>

int main()
{
    int i,sum=0;
    for (i=1;i<100;i+=2)
    {
        printf("%d ",i);
        sum=sum+i;
    }
    printf("\nSum is : %d",sum);
    return 0;
}
```
- Output Console:** Displays the execution results:

```
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47 49 51 53 5
5 57 59 61 63 65 67 69 71 73 75 77 79 81 83 85 87 89 91 93 95 97 99
Sum is : 2500
Process returned 0 (0x0)   execution time : 0.032 s
Press any key to continue.
```
- Log & Other:** Shows build messages, including "Build finished: 0 error(s), 0 warning(s) in 10 minute(s)".

}

The screenshot shows the CodeBlocks IDE with a C program open. The program is designed to print the ASCII values of characters from 'a' to 'z'. The code is as follows:

```
#include <stdio.h>
int main()
{
    int i;
    // Print ASCII values from 0 to 255
    for(i=0; i<255; i++)
    {
        printf("ASCII value of character %c = %d\n", i);
    }
    return 0;
}
```

The program is compiled and executed. The output in the console window shows the ASCII values of characters from 'a' to 'z' (ASCII values 97 to 122). The output is as follows:

```
ASCII value of character a = 97
ASCII value of character b = 98
ASCII value of character c = 99
ASCII value of character d = 100
ASCII value of character e = 101
ASCII value of character f = 102
ASCII value of character g = 103
ASCII value of character h = 104
ASCII value of character i = 105
ASCII value of character j = 106
ASCII value of character k = 107
ASCII value of character l = 108
ASCII value of character m = 109
ASCII value of character n = 110
ASCII value of character o = 111
ASCII value of character p = 112
ASCII value of character q = 113
ASCII value of character r = 114
ASCII value of character s = 115
ASCII value of character t = 116
ASCII value of character u = 117
ASCII value of character v = 118
ASCII value of character w = 119
ASCII value of character x = 120
ASCII value of character y = 121
ASCII value of character z = 122
```

The console window also shows the process return value (0) and the execution time (0.032 s).

05.

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

```
int main()
```

```
{
```

```
    int i,num,reverse=0;
```

```
    printf("Enter any number to find reverse: ");
```

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

```
    for(i=1;i<=8;i++)
```

```
    {
```

```
        reverse= (reverse*10)+(num%10);
```

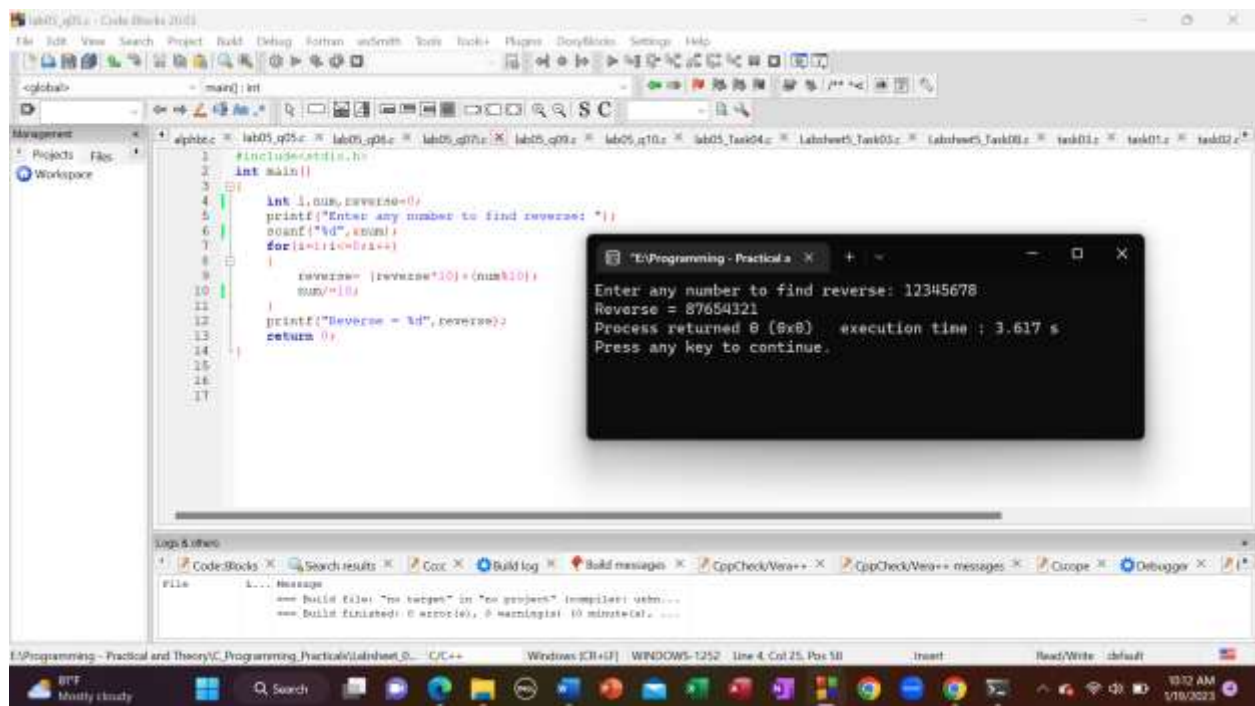
```
        num/=10;
```

```
    }
```

```
    printf("Reverse = %d",reverse);
```

```
    return 0;
```

```
}
```



The screenshot shows the Code::Blocks IDE with a C program to reverse a number. The code is as follows:

```
1 #include<stdio.h>
2 int main()
3 {
4     int i,num,reverse=0;
5     printf("Enter any number to find reverse: ");
6     scanf("%d",&num);
7     for(i=1;i<=8;i++)
8     {
9         reverse= (reverse*10)+(num%10);
10        num/=10;
11    }
12    printf("Reverse = %d",reverse);
13    return 0;
14 }
```

The output window shows the program running successfully with the input '12345678' and the output 'Reverse = 87654321'. The process returned 0 (0x0) and the execution time was 3.617 s.

06.

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

```
int main()
```

```
{
```

```
    int base,exponent,power=1,i;
```

```
    printf("Enter base : ");
```

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

```
    printf("Enter exponent : ");
```

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

```
    for(i=1;i<=exponent;i++)
```

```
    {
```

```
        power=power*base;
```

```
    }
```

```
    printf("%d ^ %d = %d",base,exponent,power);
```

```
    return 0;
```

```
}
```

```
1 #include<stdio.h>
2
3 int main()
4 {
5     int base,exponent,power=1,i;
6     printf("Enter base : ");
7     scanf("%d",&base);
8     printf("Enter exponent : ");
9     scanf("%d",&exponent);
10    for(i=1;i<=exponent;i++)
11    {
12        power=power*base;
13    }
14    printf("%d ^ %d = %d",base,exponent,power);
15    return 0;
16 }
```

Enter base : 2
Enter exponent : 3
2 ^ 3 = 8
Process returned 8 (0x0) execution time : 2.845 s
Press any key to continue.

File: I:\Programming - Practical and Theory\VC_Programming_Practicals\labSheet_06_... C/C++
Windows [C++], WINDOW5-1252 Line 16, Col 1, Pos 122
Insert Read/Write default

07.

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

```
int main()
```

```
{
```

```
    int i,c;
```

```
    printf("enter rows :");
```

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

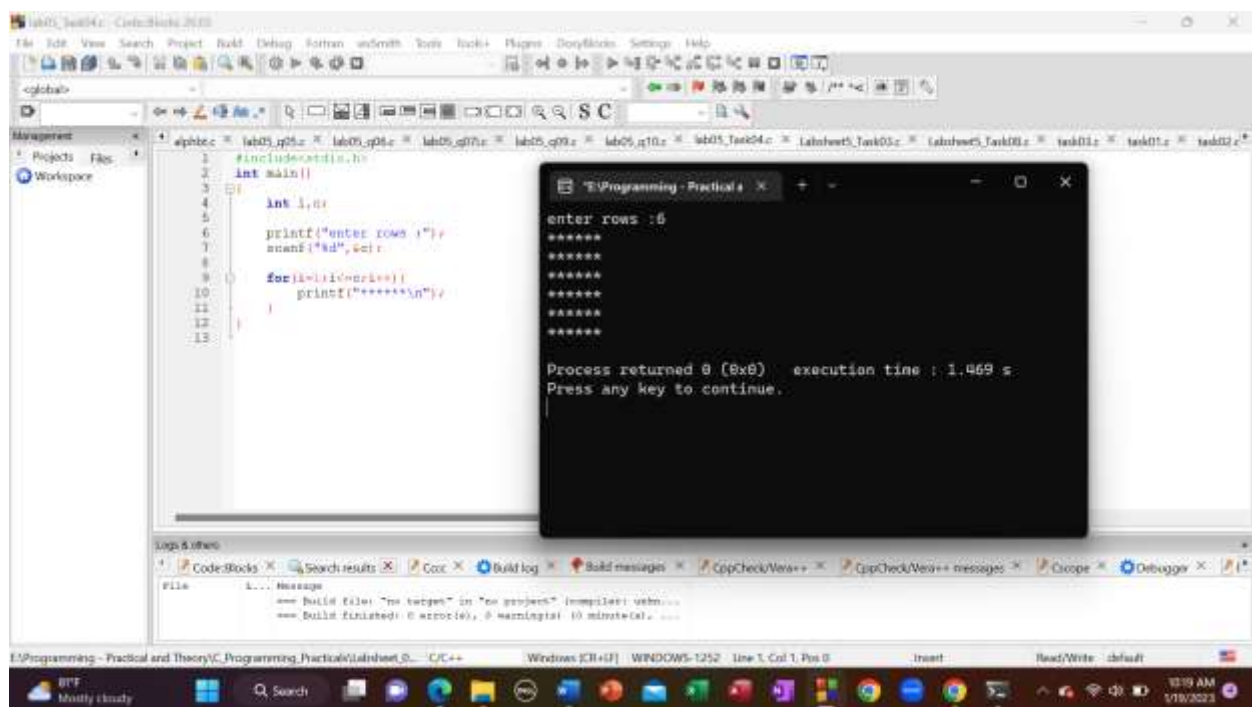
```
    for(i=1;i<=c;i++){
```

```
        printf("*****\n");
```

```
    }
```

```
    return 0;
```

```
}
```



The screenshot shows a C++ IDE with the following code in the editor:

```
1 #include<stdio.h>
2 int main()
3 {
4     int i,c;
5
6     printf("enter rows :");
7     scanf("%d",&c);
8
9     for(i=1;i<=c;i++){
10         printf("*****\n");
11     }
12
13     return 0;
14 }
```

The output window shows the execution results:

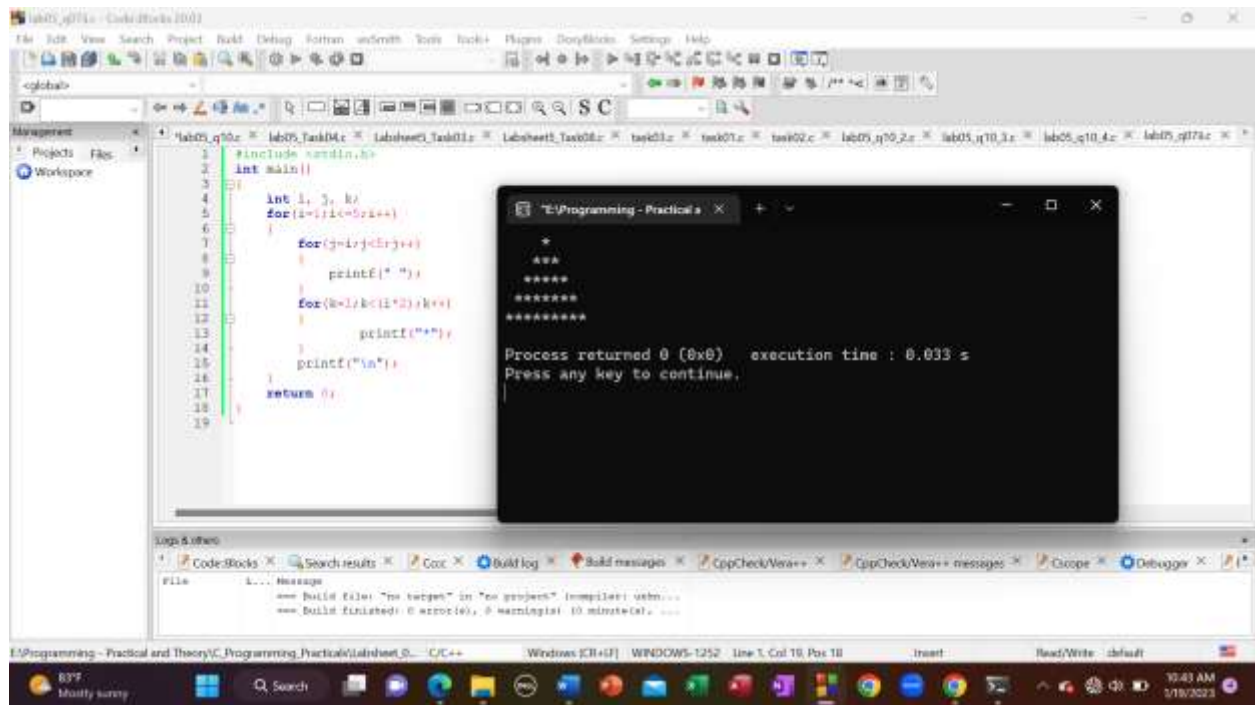
```
enter rows :6
*****
*****
*****
*****
*****
*****

Process returned 0 (0x0)   execution time : 1.469 s
Press any key to continue.
```

The status bar at the bottom indicates the file path: `I:\Programming - Practical and Theory\Programming Practical\labSheet_0... \C/C++`.


```
#include <stdio.h>

int main()
{
    int i, j, k;
    for(i=1;i<=5;i++)
    {
        for(j=i;j<=5;j++)
        {
            printf(" ");
        }
        for(k=1;k<=(i*2);k++)
        {
            printf("*");
        }
        printf("\n");
    }
    return 0;
}
```



08.

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

```
int main()
```

```
{
```

```
    int rows,i,j;
```

```
    printf("Enter no.of rows : ");
```

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

```
    for(i=1;i<=rows;i++)
```

```
    {
```

```
        for(j=1;j<=rows;j++)
```

```
        {
```

```
            if(i==1 || i==rows || j==1 || j==rows)
```

```
            {
```

```
                printf("*");
```

```
            }
```

```
        else
```

```
    {
```

```

    printf(" ");

}

}

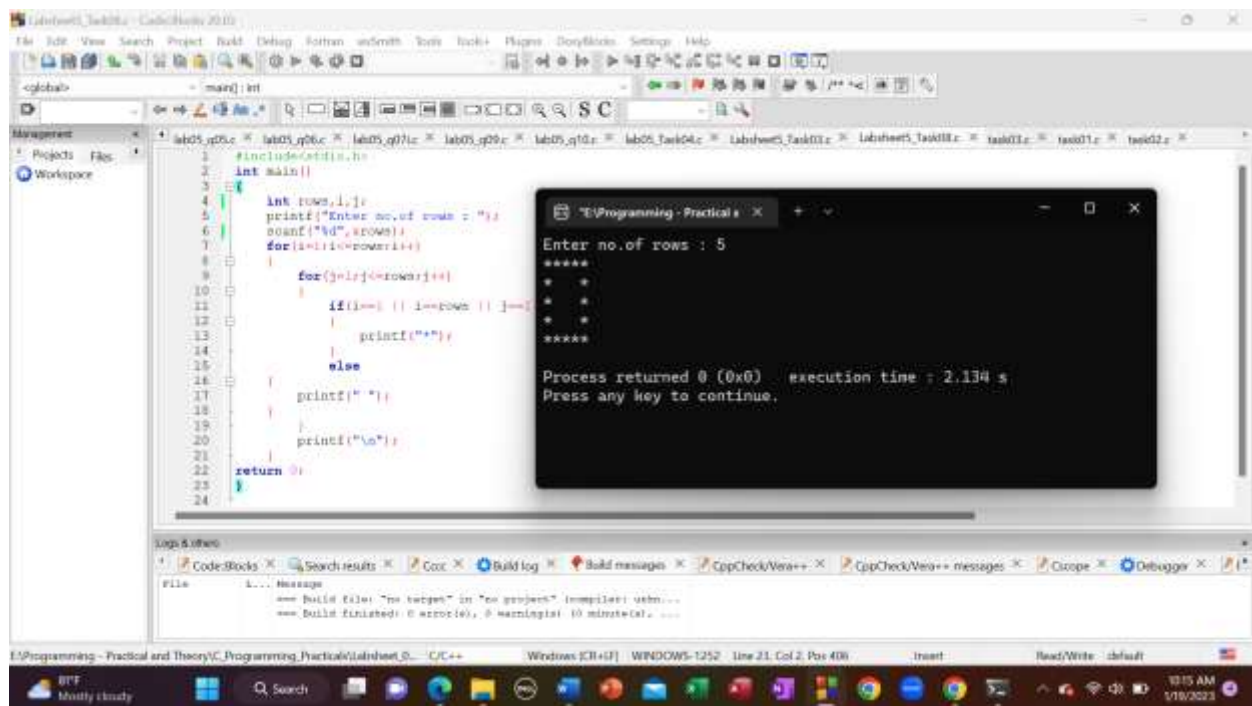
printf("\n");

}

return 0;

}

```



09.

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

```
int main()
```

```
{
```

```
    int i,c,r;
```

```
    printf("enter rows :");
```

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

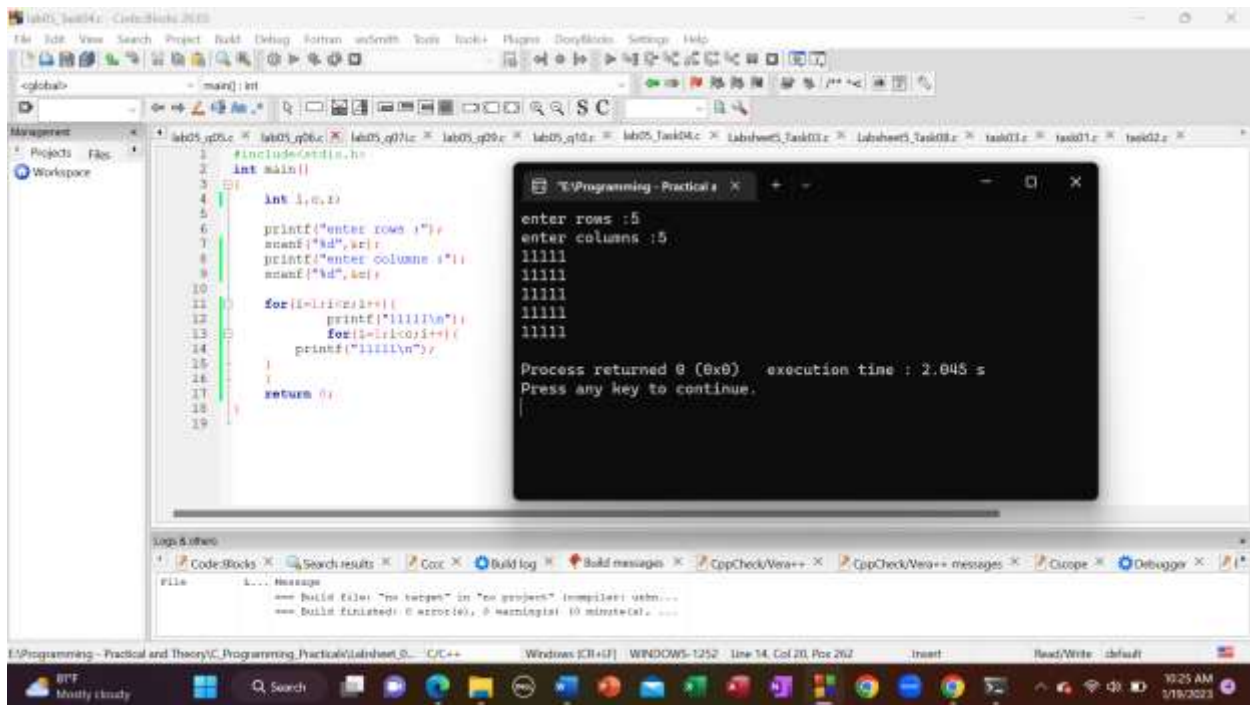
```
    printf("enter columns :");
```

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

```

for(i=1;i<r;i++){
    printf("11111\n");
    for(i=1;i<c;i++){
        printf("11111\n");
    }
}
return 0;
}

```



10.

a.

```

#include <stdio.h>

int main()
{
    int i, j;
    for(i=1;i<=5;i++)
    {

```

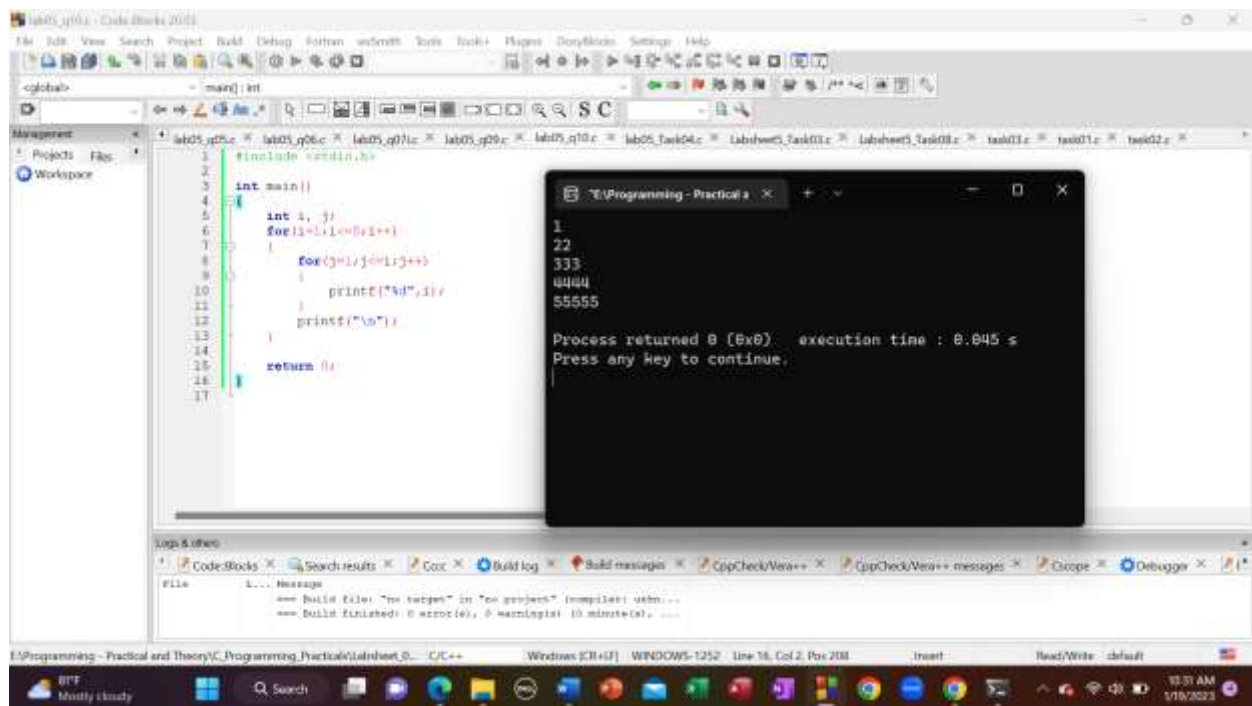
```

    for(j=1;j<=i;j++)
    {
        printf("%d",i);
    }

    printf("\n");
}

return 0;
}

```



b.

```

#include <stdio.h>

int main()
{
    int i, j;
    for(i=1;i<=5;i++)
    {
        for(j=i;j<=5;j++)
        {

```

```

        printf("%d",j);

    }

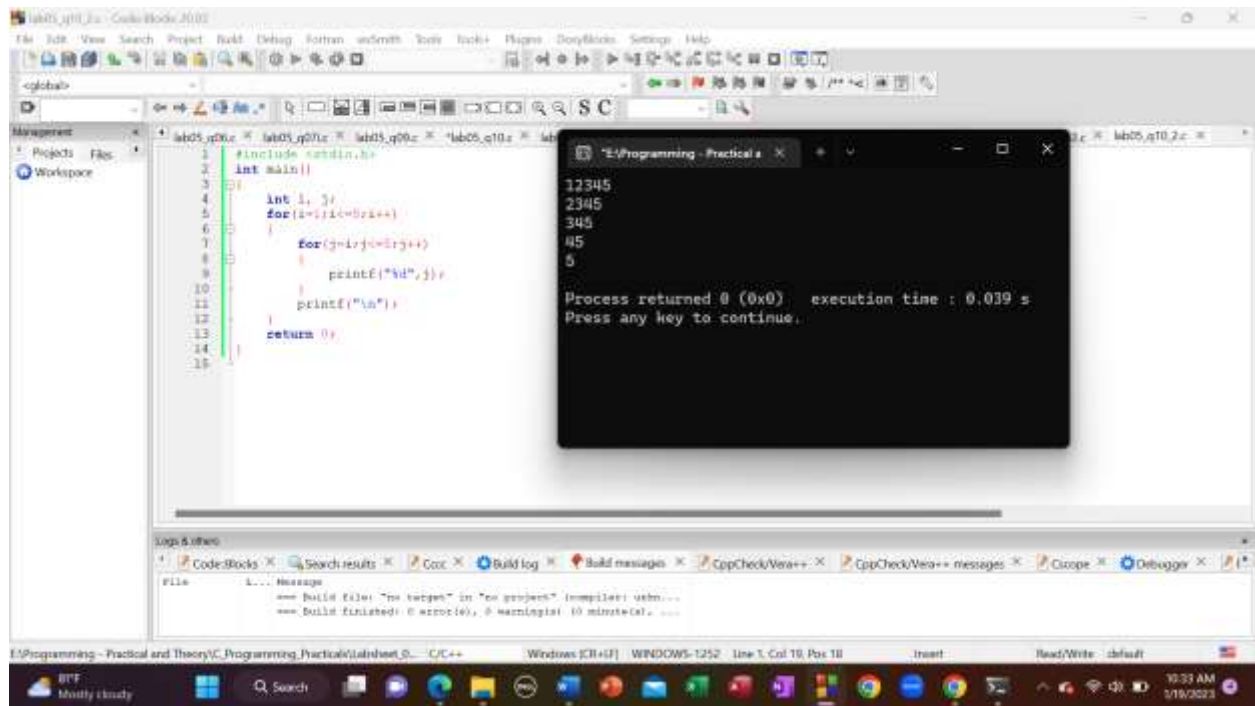
    printf("\n");

}

return 0;

}

```



c.

```

#include <stdio.h>

int main()
{
    int i, j;
    for(i=1;j<=5;i++)
    {
        for(j=5;j>=i;j--)
        {
            printf("%d",j);
        }
    }
}

```

```

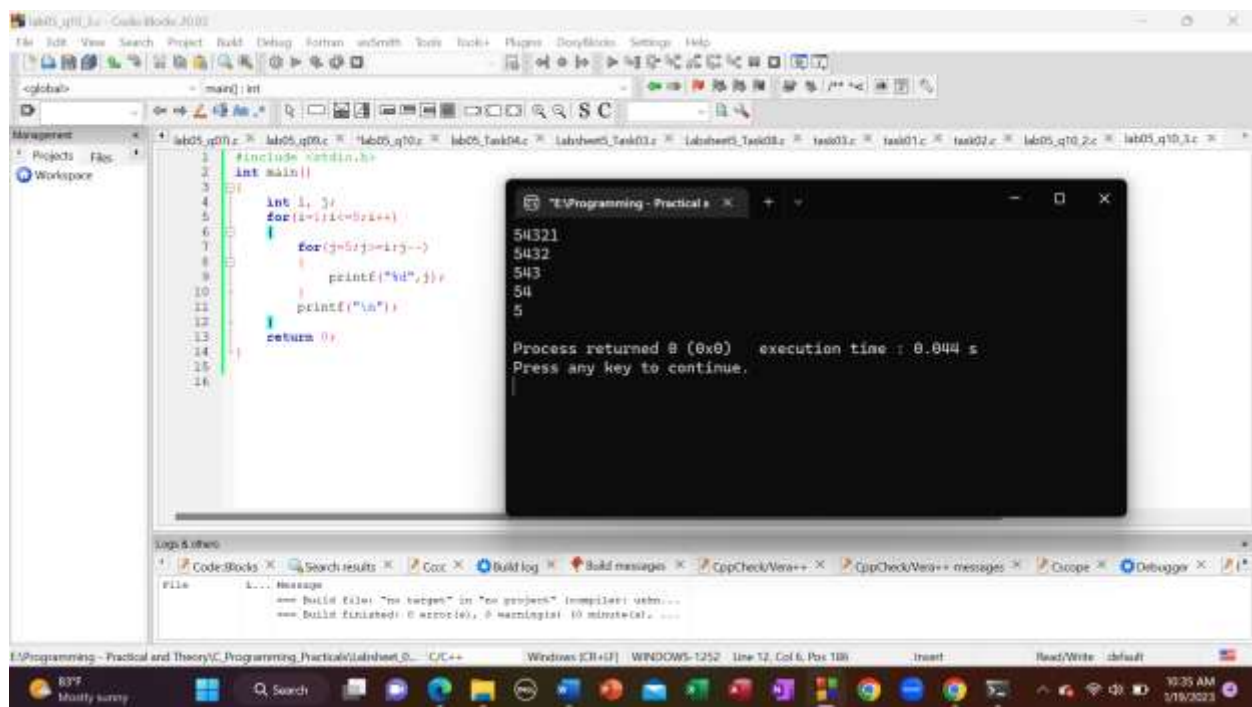
printf("\n");

}

return 0;

}

```



d.

```

#include <stdio.h>
int main()
{
    int i, j;
    for(i=1;i<=5;i++)
    {
        for(j=i;j>=1;j--)
        {
            printf("%d",j);
        }
        printf("\n");
    }
    return 0;
}

```

The screenshot shows a C++ IDE with a source file named `main.cpp`. The code implements a program that prints numbers from 1 to 54321 in a specific pattern using nested for loops. The output window displays the following sequence of numbers:

```
1
21
321
4321
54321
```

Below the output, the console shows the process return status and execution time:

```
Process returned 0 (0x0)   execution time : 0.026 s
Press any key to continue.
```

The IDE interface includes a file explorer on the left, a toolbar at the top, and a status bar at the bottom.

Discussion: We have discussed the for loop and nested for loop, and now I can code with the nested for loop and for loop statements.

References:

Cfor loop. (n.d.). W3schools.com. Retrieved March 25, 2023, from

https://www.w3schools.com/c/c_for_loop.php

Cfor Loop (with examples) – algbly. (n.d.). Algbly.com. Retrieved March 25, 2023,

from <https://www.algbly.com/Tutorials/C-programming/C-for-loop.html>