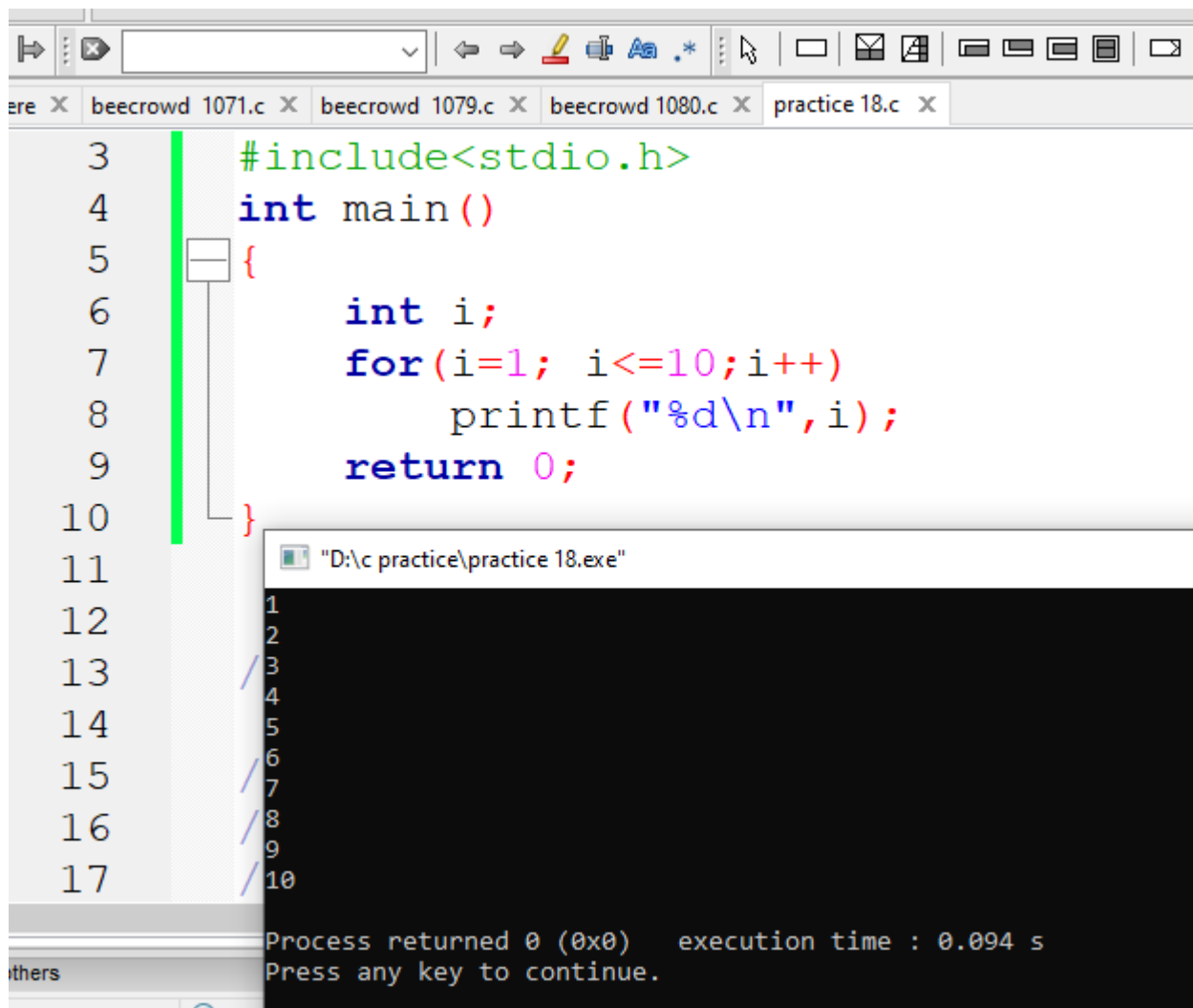


1. Write a C program to print 1-10.

```
#include<stdio.h>

int main()
{
    int i;
    for(i=1; i<=10; i++)
        printf("%d\n", i);
    return 0;
}
```



```
#include<stdio.h>
int main()
{
    int i;
    for(i=1; i<=10; i++)
        printf("%d\n", i);
    return 0;
}
```

"D:\c practice\practice 18.exe"

```
1
2
3
4
5
6
7
8
9
10
```

Process returned 0 (0x0) execution time : 0.094 s
Press any key to continue.

2. Write c Program to print 10-1

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

```
int main()
```

```
{
```

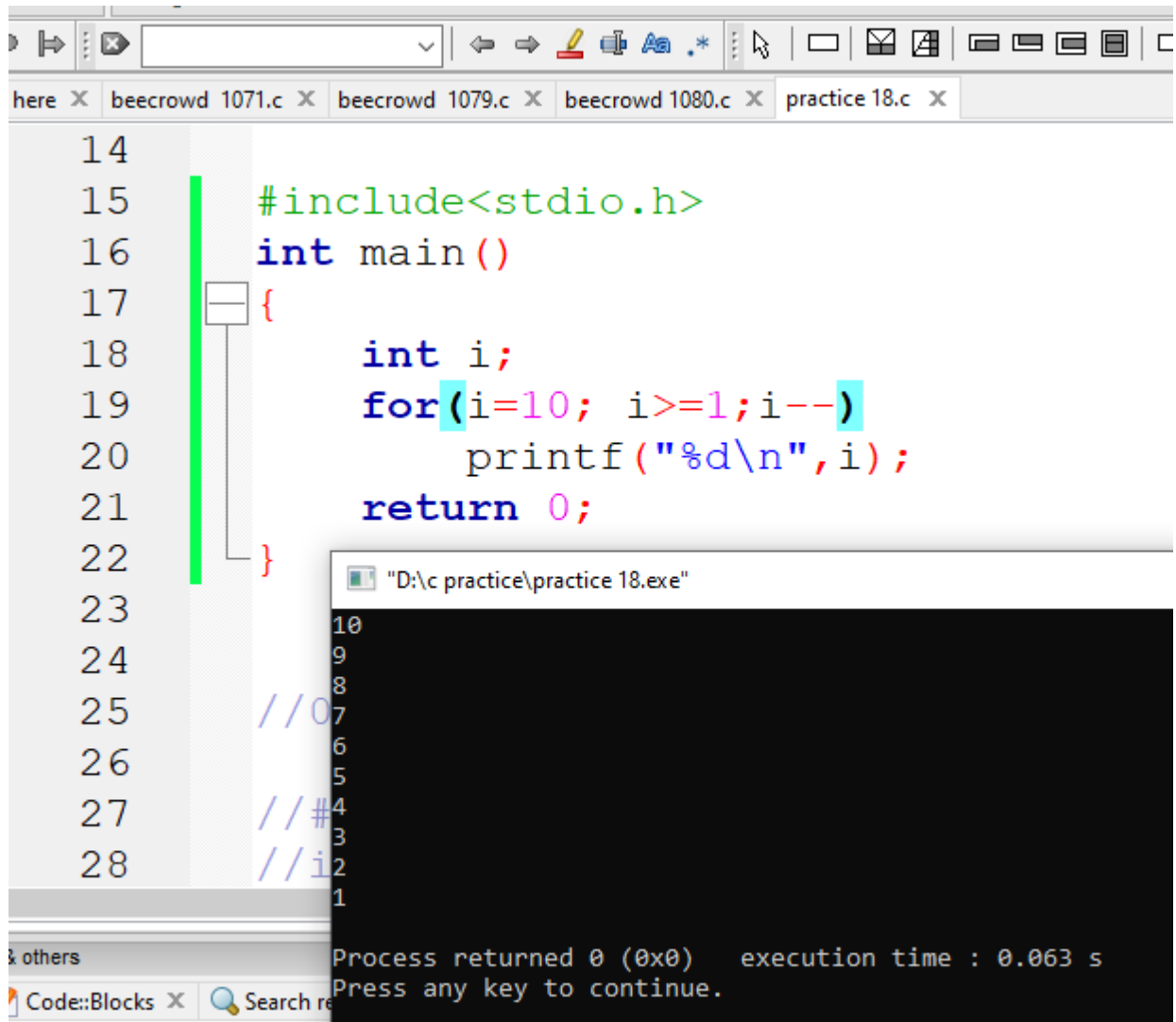
```
    int i;
```

```
    for(i=10; i>=1; i--)
```

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

```
    return 0;
```

```
}
```



The image shows a screenshot of a code editor window with multiple tabs. The active tab is 'practice 18.c'. The code in the editor is a C program to print numbers from 10 down to 1. The code is as follows:

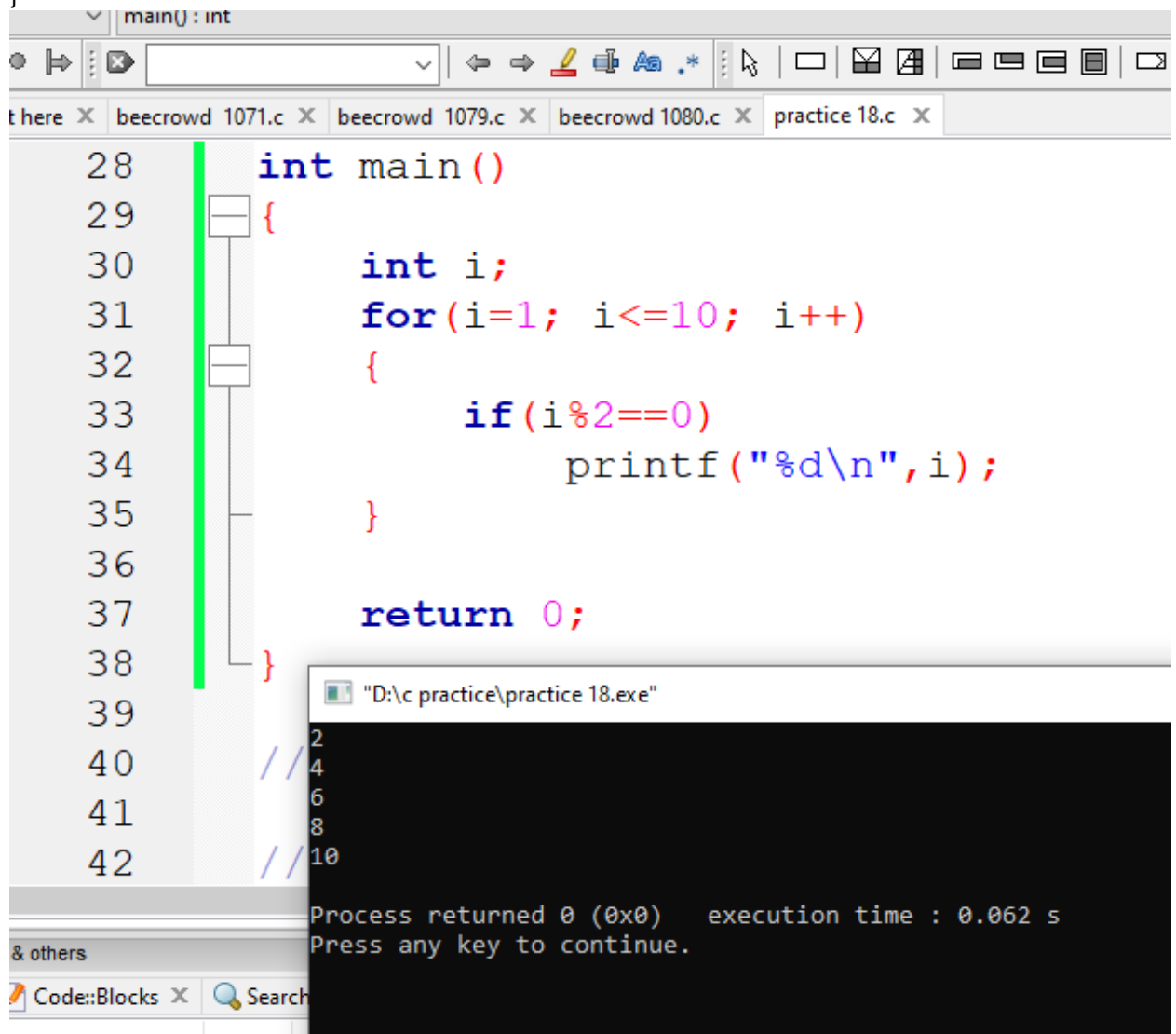
```
14
15     #include<stdio.h>
16     int main()
17     {
18         int i;
19         for(i=10; i>=1; i--)
20             printf("%d\n", i);
21         return 0;
22     }
23
24
25     //0
26
27     //4
28     //i
```

Below the code editor, there is a terminal window titled '"D:\c practice\practice 18.exe"'. The terminal displays the output of the program, which is the numbers 10 through 1, each on a new line. At the bottom of the terminal, it says 'Process returned 0 (0x0) execution time : 0.063 s' and 'Press any key to continue.'

3. Print all even numbers between 1-10.

```
#include<stdio.h>
int main()
{
    int i;
    for(i=1; i<=10; i++)
    {
        if(i%2==0)
            printf("%d\n",i);
    }

    return 0;
}
```



The screenshot shows a code editor window with the following tabs: "t here", "beecrowd 1071.c", "beecrowd 1079.c", "beecrowd 1080.c", and "practice 18.c". The code in the editor is the same as the one in the previous block. To the right of the code, there is a terminal window titled "D:\c practice\practice 18.exe" showing the output of the program: 2, 4, 6, 8, 10. Below the output, the terminal displays "Process returned 0 (0x0) execution time : 0.062 s" and "Press any key to continue.".

```
28      int main()
29      {
30          int i;
31          for(i=1; i<=10; i++)
32          {
33              if(i%2==0)
34                  printf("%d\n",i);
35          }
36
37          return 0;
38      }
39
40      //
41
42      //
```

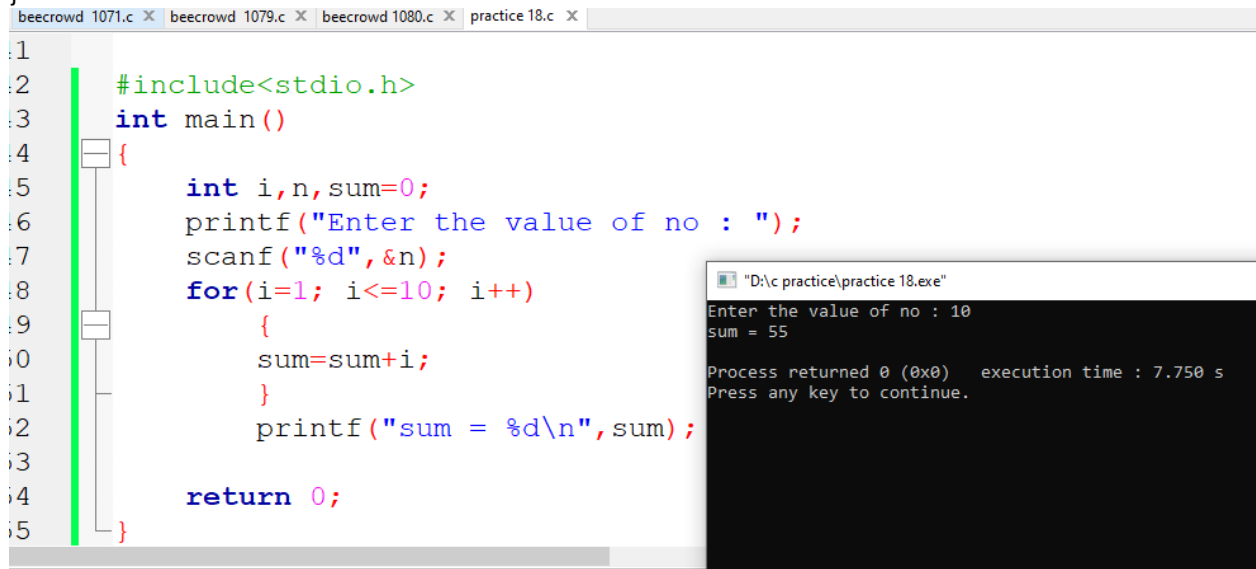
2
4
6
8
10

Process returned 0 (0x0) execution time : 0.062 s
Press any key to continue.

4. Find the sum of numbers in a given range of 1 to n. n is a number entered by the user.

```
#include<stdio.h>
int main()
{
    int i,n,sum=0;
    printf("Enter the value of no : ");
    scanf("%d",&n);
    for(i=1; i<=10; i++)
    {
        sum=sum+i;
    }
    printf("sum = %d\n",sum);

    return 0;
}
```



The screenshot shows a code editor with a file named 'practice 18.c' open. The code is a C program that calculates the sum of numbers from 1 to 10. The code is as follows:

```
1 #include<stdio.h>
2 int main()
3 {
4     int i,n,sum=0;
5     printf("Enter the value of no : ");
6     scanf("%d",&n);
7     for(i=1; i<=10; i++)
8     {
9         sum=sum+i;
10    }
11    printf("sum = %d\n",sum);
12
13    return 0;
14 }
```

Below the code editor, a terminal window titled '"D:\c practice\practice 18.exe"' shows the program's execution. The user has entered '10' for the value of 'no'. The program outputs 'sum = 55'. The terminal also displays 'Process returned 0 (0x0) execution time : 7.750 s' and 'Press any key to continue.'.

5. Find the sum of all even numbers in a range 1 to n.

```

#include<stdio.h>
int main()
{
    int i,n,sum=0;
    printf("Enter the value of no : ");
    scanf("%d",&n);
    for(i=1; i<=n; i++)
    {
        if(i%2==0)
            sum=sum+i;
    }
    printf("sum = %d\n",sum);
    return 0;
}

```

The screenshot shows a code editor with the following C program:

```

58
59 #include<stdio.h>
60 int main()
61 {
62     int i,n,sum=0;
63     printf("Enter the value of no : ");
64     scanf("%d",&n);
65     for(i=1; i<=n; i++)
66     {
67         if(i%2==0)
68             sum=sum+i;
69     }
70     printf("sum = %d\n",sum);
71     return 0;
72 }
73

```

To the right of the code editor is a terminal window titled "D:\c practice\practice 18.exe". It displays the following output:

```

Enter the value of no : 5
sum = 6

Process returned 0 (0x0)   execution time : 9.234 s
Press any key to continue.

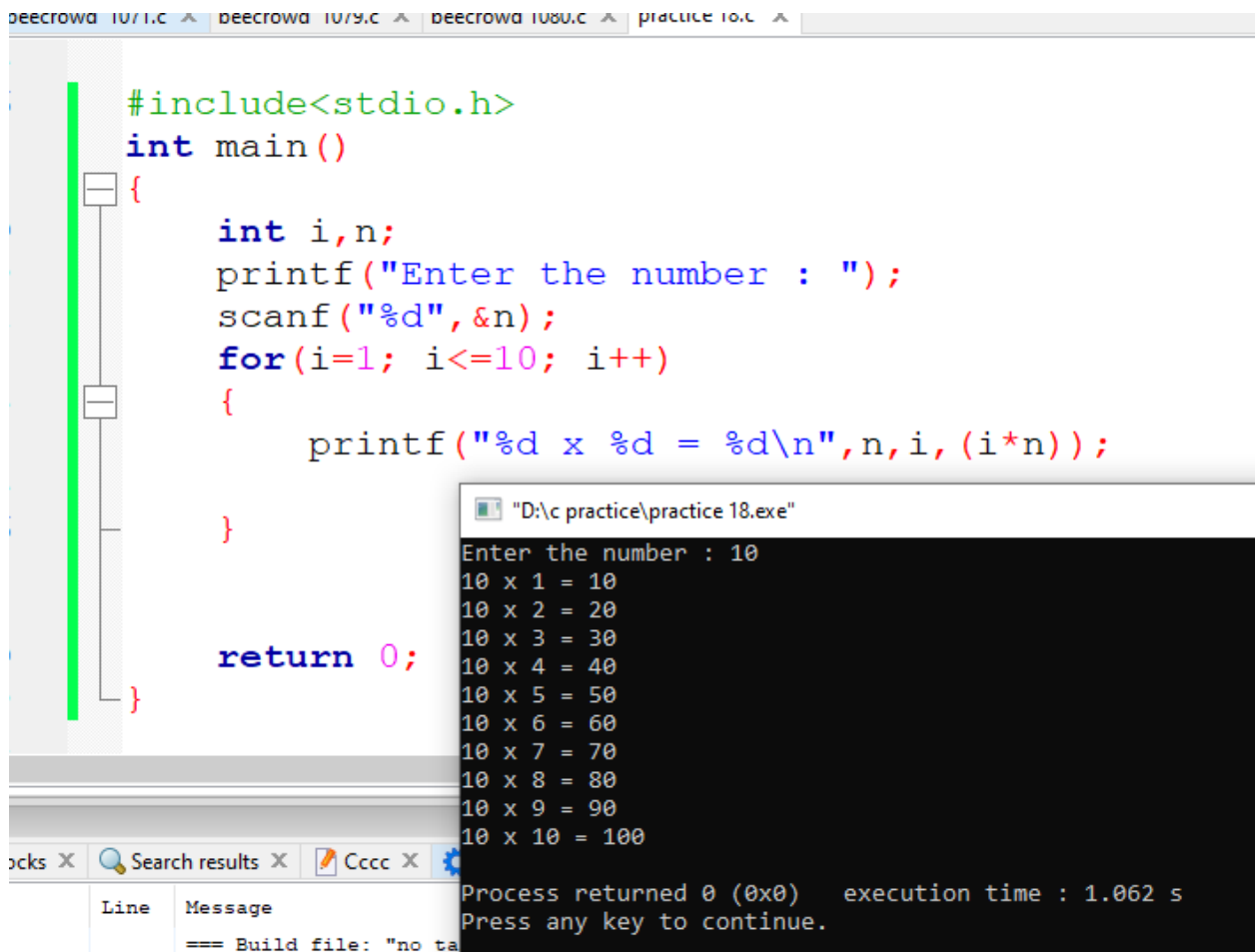
```

6. Take a number from the user and print the multiplication tables of that number.

```

#include<stdio.h>
int main()
{
    int i,n;
    printf("Enter the number : ");
    scanf("%d",&n);
    for(i=1; i<=10; i++)
    {
        printf("%d x %d = %d\n",n,i,(i*n));
    }
    return 0;
}

```



The screenshot shows a C program in a code editor with tabs for 'beecrowd 10/1.c', 'beecrowd 10/9.c', 'beecrowd 1080.c', and 'practice 18.c'. The code is as follows:

```
#include<stdio.h>
int main()
{
    int i,n;
    printf("Enter the number : ");
    scanf("%d",&n);
    for(i=1; i<=10; i++)
    {
        printf("%d x %d = %d\n",n,i,(i*n));
    }

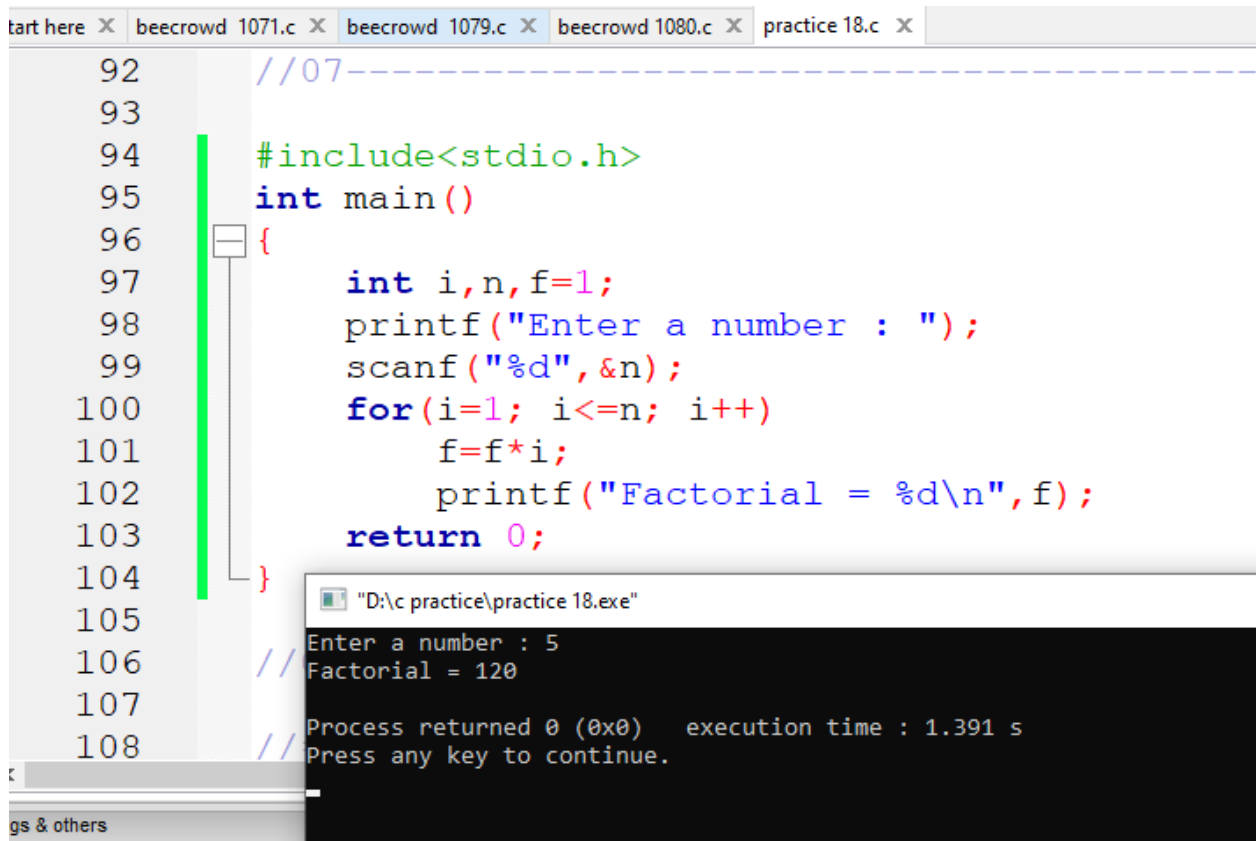
    return 0;
}
```

The console window titled "D:\c practice\practice 18.exe" shows the program's execution. It prompts "Enter the number : 10" and then displays the multiplication results for 10 multiplied by integers from 1 to 10. At the bottom, it states "Process returned 0 (0x0) execution time : 1.062 s" and "Press any key to continue."

Line	Message
==	Build file: "no ta

7. Take a number from the user and print the factorial of that number.

```
#include<stdio.h>
int main()
{
    int i,n,f=1;
    printf("Enter a number : ");
    scanf("%d",&n);
    for(i=1; i<=n; i++)
        f=f*i;
    printf("Factorial = %d\n",f);
    return 0;
}
```



The screenshot shows a C++ IDE with several tabs: 'tart here', 'beecrowd 1071.c', 'beecrowd 1079.c', 'beecrowd 1080.c', and 'practice 18.c'. The 'practice 18.c' tab is active, displaying the following code:

```
92 //07-----
93
94 #include<stdio.h>
95 int main()
96 {
97     int i,n,f=1;
98     printf("Enter a number : ");
99     scanf("%d",&n);
100     for(i=1; i<=n; i++)
101         f=f*i;
102     printf("Factorial = %d\n",f);
103     return 0;
104 }
105
106 //
107
108 //
```

Below the code editor, a console window titled '"D:\c practice\practice 18.exe"' shows the program's execution. It prompts 'Enter a number : 5' and outputs 'Factorial = 120'. At the bottom, it displays 'Process returned 0 (0x0) execution time : 1.391 s' and 'Press any key to continue.'.

8. Take a number from the user and print the proper divisor of that number.

```
#include<stdio.h>
int main()
{
    int n,i;
    printf("Enter a Number: ");
    scanf("%d",&n);
    printf("Divisor = ");
    for(i=1; i<=n; i++)
    {
        if( n%i ==0)
        {
            printf("%d,",i);
        }
    }
    printf("\n");
    return 0;
}
```

```
re x beecrowd 1071.c x beecrowd 1079.c x beecrowd 1080.c x practice 18.c x
l07
l08 #include<stdio.h>
l09 int main()
l10 {
l11     int n,i;
l12     printf("Enter a Number: ");
l13     scanf("%d",&n);
l14     printf("Divisor = ");
l15     for(i=1; i<=n; i++)
l16     {
l17         if( n%i ==0)
l18         {
l19             printf("%d,",i);
l20         }
l21     }
l22     printf("\n");
l23     return 0;
l24 }
```

9. Write a program to count the number of digits in a given number.

```
#include<stdio.h>
int main()
{
    int n,count=0;
    printf("Enter a number : ");
    scanf("%d",&n);
    while(n!=0)//if is true?
    {
        n=n/10;//continue again
        count++;
    }
    printf("Total digit : %d\n",count);
    return 0;
}
```



```
beecrowd 1071.c X beecrowd 1079.c X beecrowd 1080.c X practice 18.c X
7
3  #include<stdio.h>
9  int main()
0  {
1      int n,count=0;
2      printf("Enter a number : ");
3      scanf("%d",&n);
4      while(n!=0)//if is true?
5      {
6          n=n/10;//continue again
7          count++;
8      }
9      printf("Total digit : %d\n",count);
0      return 0;
1  }
```

```
"D:\c practice\practice 18.exe"
Enter a number : 123452
Total digit : 6
Process returned 0 (0x0)   execution time : 4.594 s
Press any key to continue.
```

10. Write a program to calculate the sum of the first and the last digits in a given number.

```
#include<stdio.h>
int main()
{
    int n,f,l,sum=0;//f=first digit l=last digit
    printf("Enter a Number:");
    scanf("%d",&n);
    l=n%10;//find the last digit
    f=n;//copy n to first digit
    while(n>=10)//if is true?
    {
        n=n/10;//continue again
        f=n;//when get first digit
        sum=f+l;//then sum
    }
    printf("sum : %d\n",sum);
    return 0;
}
```

```
147 {
148     int n, f, l, sum=0; //f=first digit l=last digit
149     printf("Enter a Number:");
150     scanf("%d", &n);
151     l=n%10; //find the last digit
152     f=n; //copy n to first digit
153     while(n>=10) //if is true?
154     {
155         n=n/10; //continue again
156         f=n; //when get first digit
157         sum=f+l; //then sum
158     }
159     printf("sum : %d\n", sum);
160     return 0;
161 }
162
163 //
164
165 //
```

"D:\c practice\practice 18.exe"

Enter a Number:403

sum : 7

Process returned 0 (0x0) execution time : 7.031 s

Press any key to continue.

11. Write a program to calculate the sum of the digits in a given number.

```
#include<stdio.h>
int main()
{
    int num,temp=0,sum=0,remainder=0;
    printf("Enter A number : ");
    scanf("%d",&num);
    temp=num;
    while(temp!=0)
    {
        remainder=temp%10;
        sum+=remainder;
        temp=temp/10;
    }
    printf("Sum : %d\n",sum);
    return 0;
}
```

...s & others

```
#include<stdio.h>

int main()
{
    int n,remainder=0,sum=0;
    printf("Enter a number : ");
    scanf("%d",&n);
    while(n>0)
    {
        remainder=n%10;
        sum=(sum*10)+remainder;
        n=n/10;
    }
    printf("Reverse = %d\n",sum);
    return 0;
}
```

```
here X beecrowd 1071.c X beecrowd 1079.c X beecrowd 1080.c X practice 18.c X
184 #include<stdio.h>
185 int main()
186 {
187     int n, remainder=0, sum=0;
188     printf("Enter a number : ");
189     scanf("%d", &n);
190     while(n>0)
191     {
192         remainder=n%10;
193         sum=(sum*10)+remainder;
194         n=n/10;
195     }
196     printf("Reverse = %d\n", sum);
197     return 0;
198 }
199
200 //
201
202 //
others
```

"D:\c practice\practice 18.exe"

Enter a number : 321
Reverse = 123

Process returned 0 (0x0) execution time : 1.283 s
Press any key to continue.

13. Write a program to check whether a given number is a palindrome or not.

```
#include<stdio.h>
int main()
{
    int num,temp=0,sum=0,remainder=0;
    printf("Enter A number : ");
    scanf("%d",&num);
    temp=num;
    while(temp!=0)
    {
        remainder=temp%10;
        sum=sum*10+remainder;
        temp=temp/10;
    }
    if(num==sum)
        printf("Palindrome\n");
    else
        printf("Not palindrome\n");
    return 0;
}
```

part here X pointer 01.c X String 01.c X practice 18.c X

```
201
202 #include<stdio.h>
203 int main()
204 {
205     int num,temp=0,sum=0,remainder=0;
206     printf("Enter A number : ");
207     scanf("%d",&num);
208     temp=num;
209     while(temp!=0)
210     {
211         remainder=temp%10;
212         sum=sum*10+remainder;
213         temp=temp/10;
214     }
215     if(num==sum)
216         printf("Palindrome\n");
217     else
218         printf("Not palindrome\n");
219     return 0;
220 }
221
```

"D:\c practice\practice 18.exe"

Enter A number : 404
Palindrome

Process returned 0 (0x0) execution time : 1.422 s
Press any key to continue.

14. Take a number from the user and print that number in words.

```
#include<stdio.h>
int main()
{
    int n,remainder=0,sum=0,x;
    printf("Enter a number : ");
    scanf("%d",&n);
    while(n>0)//for this loop number reverse. like 1022 to 2201
    {
        remainder=n%10;
        sum=(sum*10)+remainder;
        n=n/10;
    }
    while(sum>0)
    {
        x=sum%10;
        sum=sum/10;
        if(x==0)
            printf("Zero ");
        if(x==1)
            printf("One ");
        if(x==2)
            printf("Two ");
        if(x==3)
            printf("Three ");
        if(x==4)
            printf("Four ");
        if(x==5)
            printf("Five ");
        if(x==6)
            printf("Six ");
        if(x==7)
            printf("Seven ");
        if(x==8)
            printf("Eight ");
        if(x==9)
            printf("Nine ");
    }
    return 0;
}
```

The screenshot shows a C++ IDE with three tabs: 'Start here', 'pointer 01.c', and 'String 01.c'. The active tab is 'practice 18.c', which contains the following code:

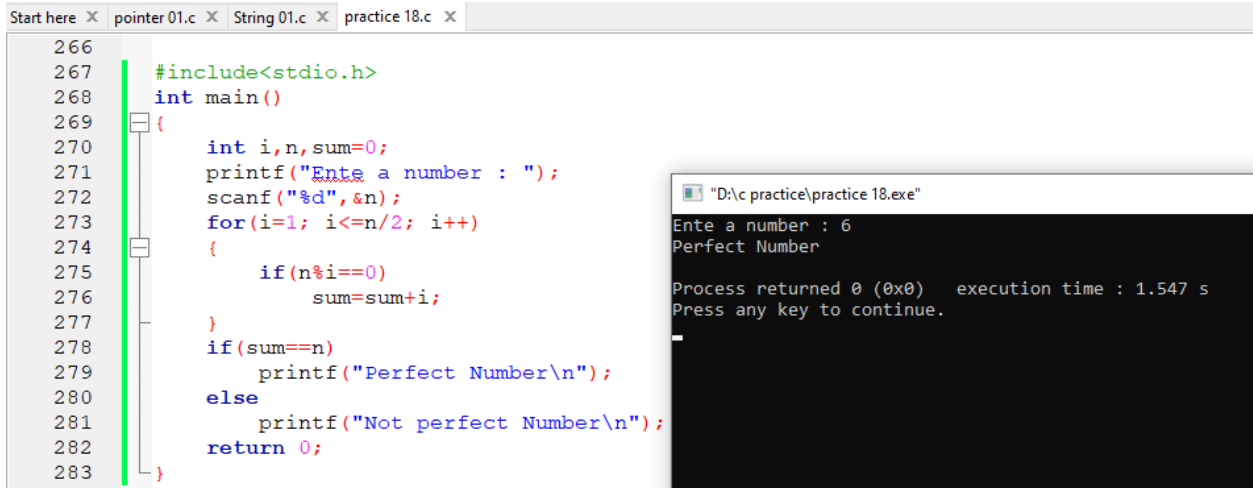
```
223
224 #include<stdio.h>
225 int main()
226 {
227     int n,remainder=0,sum=0,x;
228     printf("Enter a number : ");
229     scanf("%d",&n);
230     while(n>0)//for this loop number reverse. like 1022 to 2201
231     {
232         remainder=n%10;
233         sum=(sum*10)+remainder;
234         n=n/10;
235     }
236     while(sum>0)
237     {
238         x=sum%10;
239         sum=sum/10;
240         if(x==0)
241             printf("Zero ");
242         if(x==1)
243             printf("One ");
244         if(x==2)
245             printf("Two ");
246         if(x==3)
```

Below the code editor, a console window titled '"D:\c practice\practice 18.exe"' displays the program's output:

```
Enter a number : 1022
One Zero Two Two
Process returned 0 (0x0)   execution time : 1.719 s
Press any key to continue.
```

15. Take a number from the user and check whether it is a perfect number or not.

```
#include<stdio.h>
int main()
{
    int i,n,sum=0;
    printf("Ente a number : ");
    scanf("%d",&n);
    for(i=1; i<=n/2; i++)
    {
        if(n%i==0)
            sum=sum+i;
    }
    if(sum==n)
        printf("Perfect Number\n");
    else
        printf("Not perfect Number\n");
    return 0;
}
```

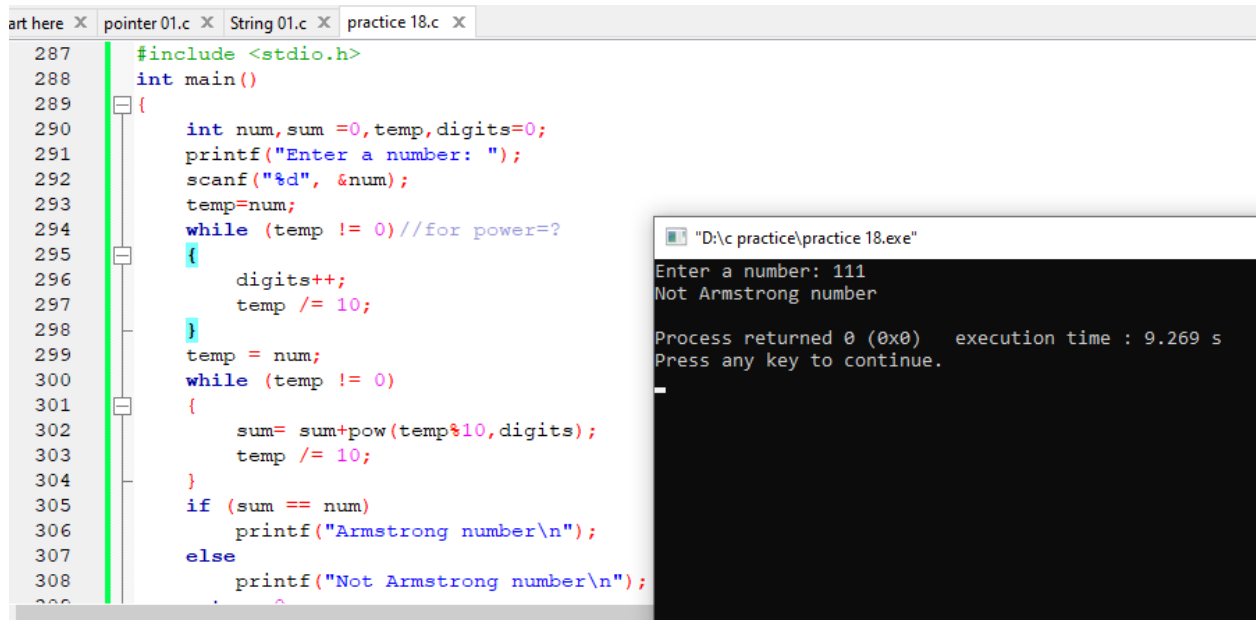
The image shows a code editor with a C program for checking perfect numbers. The code is as follows:

```
266
267 #include<stdio.h>
268 int main()
269 {
270     int i,n,sum=0;
271     printf("Enter a number : ");
272     scanf("%d",&n);
273     for(i=1; i<=n/2; i++)
274     {
275         if(n%i==0)
276             sum=sum+i;
277     }
278     if(sum==n)
279         printf("Perfect Number\n");
280     else
281         printf("Not perfect Number\n");
282     return 0;
283 }
```

Next to the code is a terminal window titled "D:\c practice\practice 18.exe". It shows the program's execution: it prompts "Enter a number : 6", outputs "Perfect Number", and then displays "Process returned 0 (0x0) execution time : 1.547 s" and "Press any key to continue.".

16. Take a number from the user and check whether it is an armstrong number or not.

```
#include <stdio.h>
int main()
{
    int num,sum=0,temp,digits=0;
    printf("Enter a number: ");
    scanf("%d", &num);
    temp=num;
    while (temp != 0)//for power=?
    {
        digits++;
        temp /= 10;
    }
    temp = num;
    while (temp != 0)
    {
        sum= sum+pow(temp%10,digits);
        temp /= 10;
    }
    if (sum == num)
        printf("Armstrong number\n");
    else
        printf("Not Armstrong number\n");
    return 0;
}
```


The image shows a code editor with a C program to check if a number is an Armstrong number. The code is as follows:

```
287 #include <stdio.h>
288 int main()
289 {
290     int num, sum = 0, temp, digits = 0;
291     printf("Enter a number: ");
292     scanf("%d", &num);
293     temp = num;
294     while (temp != 0) // for power=?
295     {
296         digits++;
297         temp /= 10;
298     }
299     temp = num;
300     while (temp != 0)
301     {
302         sum = sum + pow(temp % 10, digits);
303         temp /= 10;
304     }
305     if (sum == num)
306         printf("Armstrong number\n");
307     else
308         printf("Not Armstrong number\n");
309 }
```

The execution window on the right shows the program running with the input 111, resulting in "Not Armstrong number". It also displays the process return code as 0 (0x0) and an execution time of 9.269 s.

17. Take a number from the user and check if the number is a prime number or not.

```
#include<stdio.h>
int main()
{
    int n,i,count=0;
    printf("Enter a number : ");
    scanf("%d",&n);
    for(i=2; i<n; i++)
    {
        if(n%i==0)
            count++;
    }
    if(count==0)
        printf("Prime Number\n");
    else
        printf("Not Prime Number\n");
    return 0;
}
```

```
Start here X pointer 01.c X String 01.c X practice 18.c X
314 #include<stdio.h>
315 int main()
316 {
317     int n,i,count=0;
318     printf("Enter a number : ");
319     scanf("%d",&n);
320     for(i=2; i<n; i++)
321     {
322         if(n%i==0)
323             count++;
324     }
325     if(count==0)
326         printf("Prime Number\n");
327     else
328         printf("Not Prime Number\n");
329     return 0;
330 }
331
```

```
"D:\c practice\practice 18.exe"
Enter a number : 7
Prime Number

Process returned 0 (0x0)   execution time : 1.719 s
Press any key to continue.
```

20. Write a C program to print the following shape of the star [use nested loop].

```
#include<stdio.h>
int main()
{
    int n,i,j;
    printf("Enter a number :");
    scanf("%d",&n);
    for(i=1; i<=n; i++)
    {
        for(j=1; j<=n; j++)
        {
            printf("* ");
        }
        printf("\n");
    }
    return 0;
}
```

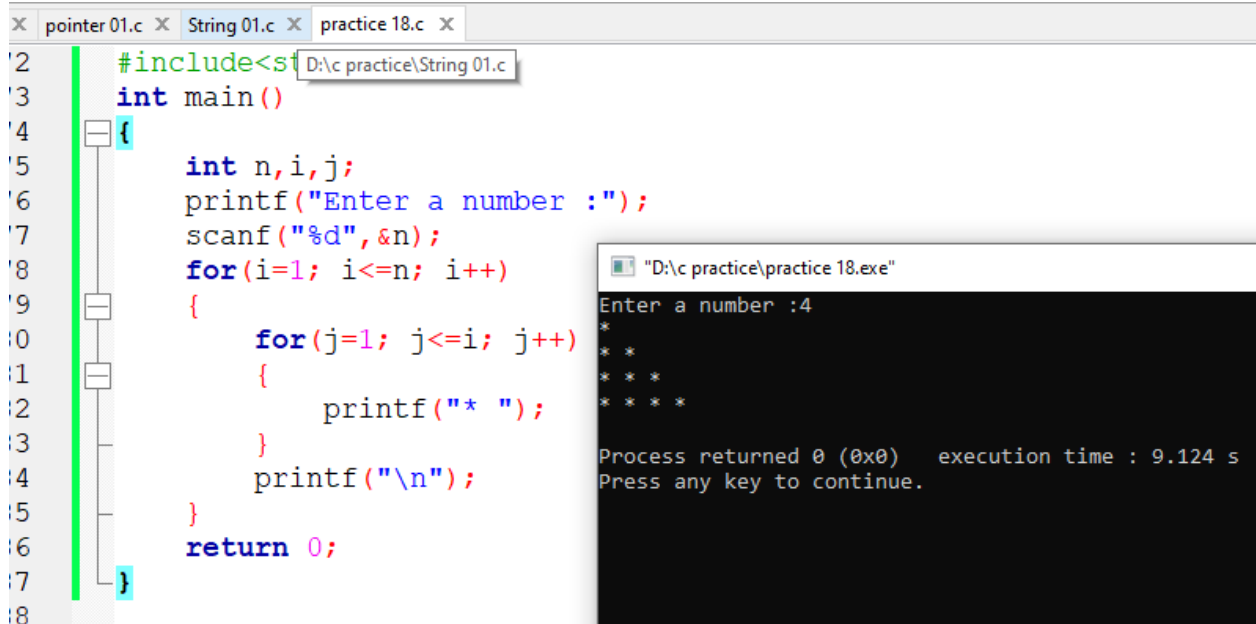
```
Start here X pointer 01.c X String 01.c X practice 18.c X
351 //20-----
352
353 #include<stdio.h>
354 int main()
355 {
356     int n,i,j;
357     printf("Enter a number :");
358     scanf("%d",&n);
359     for(i=1; i<=n; i++)
360     {
361         for(j=1; j<=n; j++)
362         {
363             printf("* ");
364         }
365         printf("\n");
366     }
367     return 0;
368 }
```

```
"D:\c practice\practice 18.exe"
Enter a number :4
* * * *
* * * *
* * * *
* * * *

Process returned 0 (0x0)   execution time : 0.719 s
Press any key to continue.
```

21. Write a C program to print the following shape of the star[use nested error]

```
#include<stdio.h>
int main()
{
    int n,i,j;
    printf("Enter a number :");
    scanf("%d",&n);
    for(i=1; i<=n; i++)
    {
        for(j=1; j<=i; j++)
        {
            printf("* ");
        }
        printf("\n");
    }
    return 0;
}
```



The screenshot shows a C program editor with three tabs: 'pointer 01.c', 'String 01.c', and 'practice 18.c'. The 'practice 18.c' tab is active, displaying the following code:

```
12 #include<stdio.h>
13 int main()
14 {
15     int n,i,j;
16     printf("Enter a number :");
17     scanf("%d",&n);
18     for(i=1; i<=n; i++)
19     {
20         for(j=1; j<=i; j++)
21         {
22             printf("* ");
23         }
24         printf("\n");
25     }
26     return 0;
27 }
```

To the right of the code editor, a terminal window titled '"D:\c practice\practice 18.exe"' shows the program's execution. It prompts 'Enter a number :4' and displays the following star pattern:

```
*
* *
* * *
* * * *
```

Below the pattern, the terminal shows the message: 'Process returned 0 (0x0) execution time : 9.124 s Press any key to continue.'

22. Write a C program to print the following shape of the star[use nested error]

```
#include<stdio.h>
int main()
{
    int n,r,c;
    printf("Enter a number : ");
    scanf("%d",&n);

    for(r=1; r<=n; r++)
    {
        for(c=1; c<=n-r; c++)
        {
            printf(" ");
        }
        for(c=1; c<=r; c++)
        {
            printf("*");
        }
        printf("\n");
    }
    return 0;
}
```

here X pointer 01.c X String 01.c X practice 18.c X

```
391 #include<stdio.h>
392 int main()
393 {
394     int n,r,c;
395     printf("Enter a number : ");
396     scanf("%d",&n);
397
398     for(r=1; r<=n; r++)
399     {
400         for(c=1; c<=n-r; c++)
401         {
402             printf(" ");
403         }
404         for(c=1; c<=r; c++)
405         {
406             printf("*");
407         }
408         printf("\n");
409     }
410     return 0;
411 }
```

"D:\c practice\practice 18.exe"

Enter a number : 4

*

**

Process returned 0 (0x0) execution time : 1.906 s
Press any key to continue.

23. Write a C program to print the following shape of the star[use nested error]

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

```
int main()
```

```
{
```

```
    int n,r,c;
```

```
    printf("Enter a number : ");
```

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

```
    for(r=1; r<=n; r++)
```

```
    {
```

```
        for(c=1; c<=2*n-1; c++)
```

```
        {
```

```
            if(c>=n-(r-1) && c<=n+(r-1))
```

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

```
            else
```

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

```
        }
```

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

```
    }
```

```
    return 0;
```

```
}
```

here X pointer 01.c X String 01.c X practice 18.c X

```
417 {
418     int n,r,c;
419     printf("Enter a number : ");
420     scanf("%d",&n);
421
422     for(r=1; r<=n; r++)
423     {
424         for(c=1; c<=2*n-1; c++)
425         {
426             if(c>=n-(r-1) && c<=n+(r-1))
427             printf("*");
428             else
429             printf(" ");
430         }
431         printf("\n");
432     }
433     return 0;
434 }
435
```

"D:\c practice\practice 18.exe"

Enter a number : 3

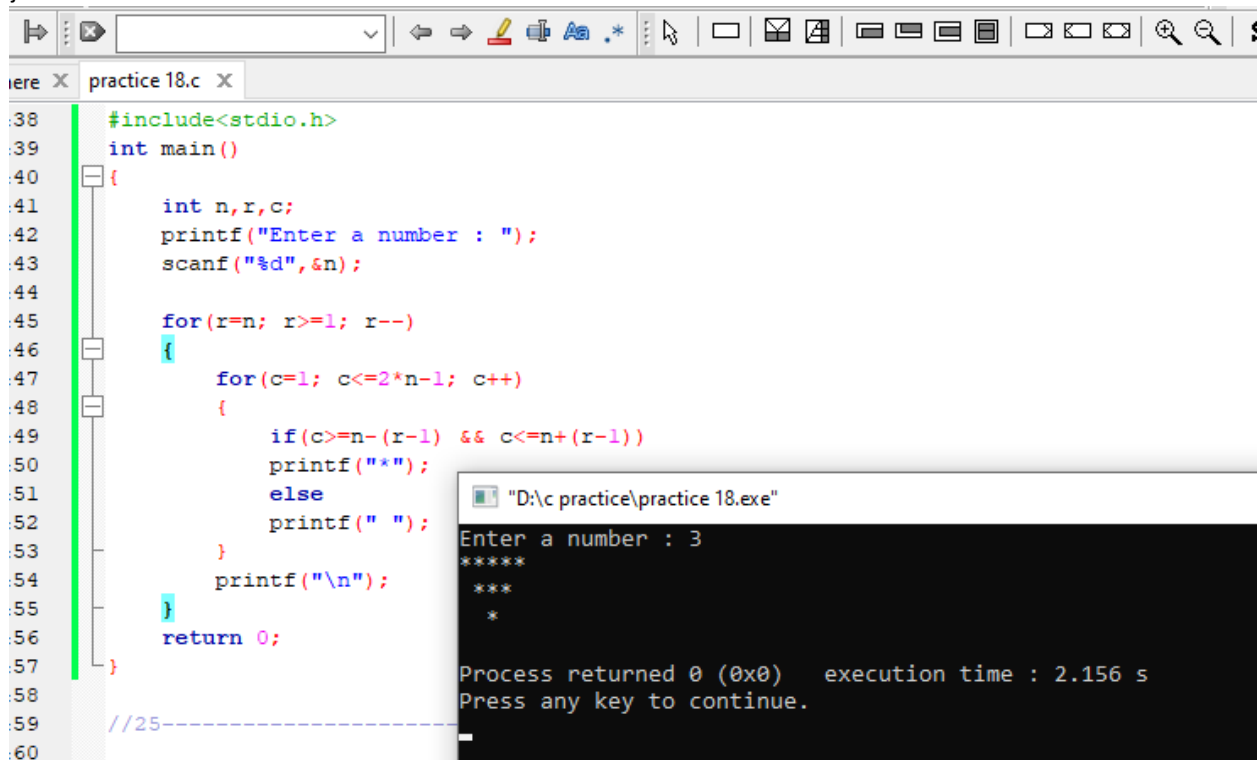
*

Process returned 0 (0x0) execution time : 0.703 s
Press any key to continue.

24. Write a C program to print the following shape of the star[use nested error]

```
#include<stdio.h>
int main()
{
    int n,r,c;
    printf("Enter a number : ");
    scanf("%d",&n);

    for(r=n; r>=1; r--)
    {
        for(c=1; c<=2*n-1; c++)
        {
            if(c>=n-(r-1) && c<=n+(r-1))
                printf("*");
            else
                printf(" ");
        }
        printf("\n");
    }
    return 0;
}
```



The screenshot shows a code editor window with a file named 'practice 18.c'. The code is a C program that prints a star pattern. The code is as follows:

```
.38 #include<stdio.h>
.39 int main()
.40 {
.41     int n,r,c;
.42     printf("Enter a number : ");
.43     scanf("%d",&n);
.44
.45     for(r=n; r>=1; r--)
.46     {
.47         for(c=1; c<=2*n-1; c++)
.48         {
.49             if(c>=n-(r-1) && c<=n+(r-1))
.50                 printf("*");
.51             else
.52                 printf(" ");
.53         }
.54         printf("\n");
.55     }
.56     return 0;
.57 }
.58
.59 //25-----
.60
```

The execution output is shown in a separate window titled "D:\c practice\practice 18.exe". It displays the prompt "Enter a number : 3" and the resulting star pattern:

```
*****
***
*
```

Below the pattern, the output shows "Process returned 0 (0x0) execution time : 2.156 s" and "Press any key to continue.".

25. Write a C program to print the following shape of the star[use nested error]

```
#include<stdio.h>
int main()
{
    int n,r,c;
    printf("Enter a number : ");
    scanf("%d",&n);

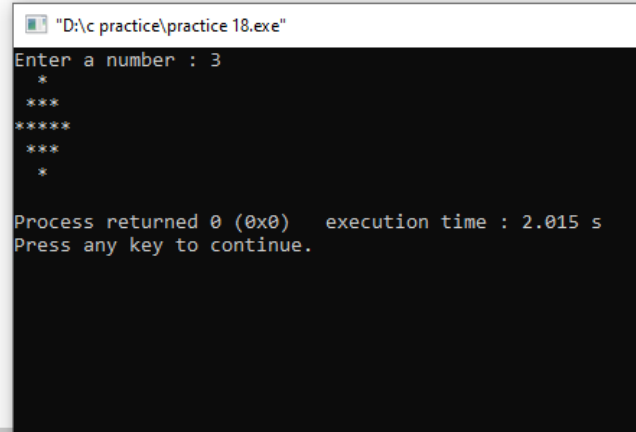
    for(r=1; r<=n; r++)
    {
        for(c=1; c<=2*n-1; c++)
        {
            if(c>=n-(r-1) && c<=n+(r-1))
                printf("*");
            else
                printf(" ");
        }
        printf("\n");
    }
    for(r=n-1; r>=1; r--)
    {
        for(c=1; c<=2*n-1; c++)
        {
            if(c>=n-(r-1) && c<=n+(r-1))
                printf("*");
            else
                printf(" ");
        }
        printf("\n");
    }
    return 0;
```

```

}

Start here x practice 18.c x
460
461 #include<stdio.h>
462 int main()
463 {
464     int n,r,c;
465     printf("Enter a number : ");
466     scanf("%d",&n);
467
468     for(r=1; r<=n; r++)
469     {
470         for(c=1; c<=2*n-1; c++)
471         {
472             if(c>=n-(r-1) && c<=n+(r-1))
473                 printf("*");
474             else
475                 printf(" ");
476         }
477         printf("\n");
478     }
479     for(r=n-1; r>=1; r--)
480     {
481         for(c=1; c<=2*n-1; c++)
482         {
483             if(c>=n-(r-1) && c<=n+(r-1))

```

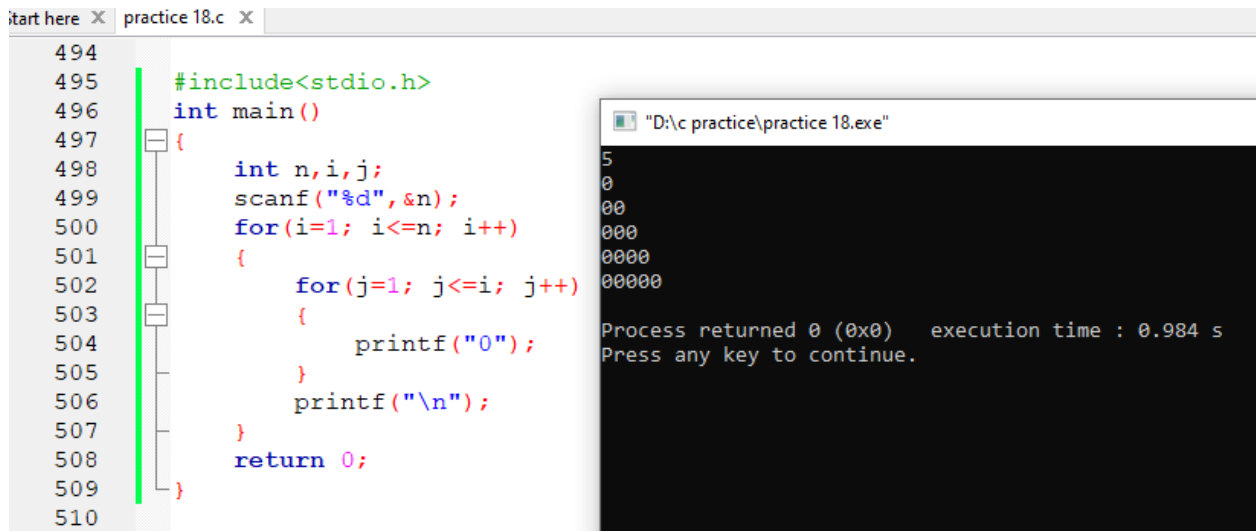


26. Write a c-program to print the following shape. But in this program you can use only one loop.

```

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

```


The image shows a code editor window with a file named 'practice 18.c' and a separate console window titled 'D:\c practice\practice 18.exe'. The code in the editor is a C program that takes an integer 'n' as input and prints a right-angled triangle of zeros. The console window shows the output for n=5, displaying five rows of zeros: '5', '0', '00', '000', and '0000'. Below the output, it states 'Process returned 0 (0x0) execution time : 0.984 s' and 'Press any key to continue.'

```
494
495 #include<stdio.h>
496 int main()
497 {
498     int n,i,j;
499     scanf("%d",&n);
500     for(i=1; i<=n; i++)
501     {
502         for(j=1; j<=i; j++)
503         {
504             printf("0");
505         }
506         printf("\n");
507     }
508     return 0;
509 }
510
```

```
5
0
00
000
0000

Process returned 0 (0x0)   execution time : 0.984 s
Press any key to continue.
```

27. Write a c-program to print the following shape. But in this program you can use only one loop.

```
#include<stdio.h>
int main()
{
    int n,r,c;
    scanf("%d",&n);

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

```
ere X practice 18.c X
512
513 #include<stdio.h>
514 int main()
515 {
516     int n,r,c;
517     scanf("%d",&n);
518
519     for(r=1; r<=n; r++)
520     {
521         for(c=1; c<=n-r; c++)
522         {
523             printf(" ");
524         }
525         for(c=1; c<=r; c++)
526         {
527             printf("0");
528         }
529         printf("\n");
530     }
531     return 0;
532 }
```

```
"D:\c practice\practice 18.exe"
5
0
00
000
0000
00000
Process returned 0 (0x0)   execution time : 0.906 s
Press any key to continue.
```

28. Write a c-program to print the following shape. But in this program you can use only one loop.

```
#include<stdio.h>
int main()
{
    int n,i,j;
    scanf("%d",&n);

    for(i=1;i<=n;i++)
    {
        for(j=1;j<=n;j++)
        {
            printf("%d",j);
        }
        printf("\n");
    }
    printf("\n\n");
    for(i=1;i<=n;i++)
    {
        for(j=1;j<=n;j++)
        {
            printf("%d",i);
        }
        printf("\n"); }
    return 0;
```

```
}
Start here x practice 18.c x
535
536 #include<stdio.h>
537 int main()
538 {
539     int n,i,j;
540     scanf("%d",&n);
541
542     for(i=1;i<=n;i++)
543     {
544         for(j=1;j<=n;j++)
545         {
546             printf("%d",j);
547         }
548         printf("\n");
549     }
550     printf("\n\n");
551     for(i=1;i<=n;i++)
552     {
553         for(j=1;j<=n;j++)
554         {
555             printf("%d",i);
556         }
557     }
558 }
```

```
"D:\c practice\practice 18.exe"
4
1234
1234
1234
1234
1111
2222
3333
4444
Process returned 0 (0x0)   execution time : 2.156 s
Press any key to continue.
```

29. Write a c-program to print the following shape. But in this program you can use only one loop.

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

```

62 }
63 #include<stdio.h>
64 int main()
65 {
66     int n,i,j;
67     scanf("%d",&n);
68     for(i=1; i<=n; i++)
69     {
70         for(j=1; j<=i; j++)
71         {
72             printf("%d",j);
73         }
74         printf("\n");
75     }
76     return 0;
77 }

```

```

"D:\c practice\practice 18.exe"
5
1
12
123
1234
12345
Process returned 0 (0x0)   execution time : 1.219 s
Press any key to continue.

```

30. Write a c-program to print the following shape. But in this program you can use only one loop.

```

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

```

```

580
581 #include<stdio.h>
582 int main()
583 {
584     int n,i,j;
585     scanf("%d",&n);
586     for(i=n; i>=1; i--)
587     {
588         for(j=1; j<=i; j++)
589         {
590             printf("%d",j);
591         }
592         printf("\n");
593     }
594     return 0;
595 }

```

```

"D:\c practice\practice 18.exe"
4
1234
123
12
1
Process returned 0 (0x0)   execution time : 1.281 s
Press any key to continue.

```

32. Write a program in C to print the Floyd's Triangle.

```
#include<stdio.h>
int main()
{
    int n,i,j,x=0;
    scanf("%d",&n);
    for(i=1; i<=n; i++)
    {
        printf("\n");
        for(j=1; j<=i; j++)
        {
            x=i-j+1;
            x=x%2;
            printf("%d",x);
        }
    }
    return 0;
}
```

```
598
599     #include<stdio.h>
600     int main()
601     {
602         int n,i,j,x=0;
603         scanf("%d",&n);
604         for(i=1; i<=n; i++)
605         {
606             printf("\n");
607             for(j=1; j<=i; j++)
608             {
609                 x=i-j+1;
610                 x=x%2;
611                 printf("%d",x);
612             }
613         }
614         return 0;
615     }
```

"D:\c practice\practice 18.exe"

```
5
1
01
101
0101
10101
Process returned 0 (0x0)   execution time : 1.692 s
Press any key to continue.
```

37. Write a C program to print the following shape of the star[use nested error.

```

#include<stdio.h>
int main()
{
    int n,r,c;
    printf("Enter a number : ");
    scanf("%d",&n);
    for(r=n; r>=1; r--)
    {
        for(c=1; c<=2*n-1; c++)
        {
            if(c>=n-(r-1) && c<=n+(r-1))
                printf("@");
            else
                printf(" ");
        }
        printf("\n");
    }
    for(r=1; r<=n; r++)
    {
        for(c=1; c<=2*n-1; c++)
        {
            if(c>=n-(r-1) && c<=n+(r-1))
                printf("@");
            else
                printf(" ");
        }
        printf("\n");
    }
    return 0;
}

```

The screenshot shows a C program being compiled and executed. The left window displays the source code with line numbers 620 to 640. The right window shows the program's output, which is a diamond shape made of '@' characters for n=5. The output also shows the process return code and execution time.

```

620
621 #include<stdio.h>
622 int main()
623 {
624     int n,r,c;
625     printf("Enter a number : ");
626     scanf("%d",&n);
627     for(r=n; r>=1; r--)
628     {
629         for(c=1; c<=2*n-1; c++)
630         {
631             if(c>=n-(r-1) && c<=n+(r-1))
632                 printf("@");
633             else
634                 printf(" ");
635         }
636         printf("\n");
637     }
638     for(r=1; r<=n; r++)
639     {
640         for(c=1; c<=2*n-1; c++)

```

Output:

```

Enter a number : 5
@@@@@@@@
@@@@@@@
@@@@@
@@@@
@@@
@@
@
@@@
@@@@
@@@@@@
@@@@@@@@

Process returned 0 (0x0)   execution time : 1.406 s
Press any key to continue.

```

40. Write a program to calculate the sum of numbers in a range from 1-20. Then check whether that sum is even or odd

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

```
int main()
```

```
{
```

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

```
    for(i=1; i<=20;i++)
```

```
    {
```

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

```
    }
```

```
    if(sum%2==0)
```

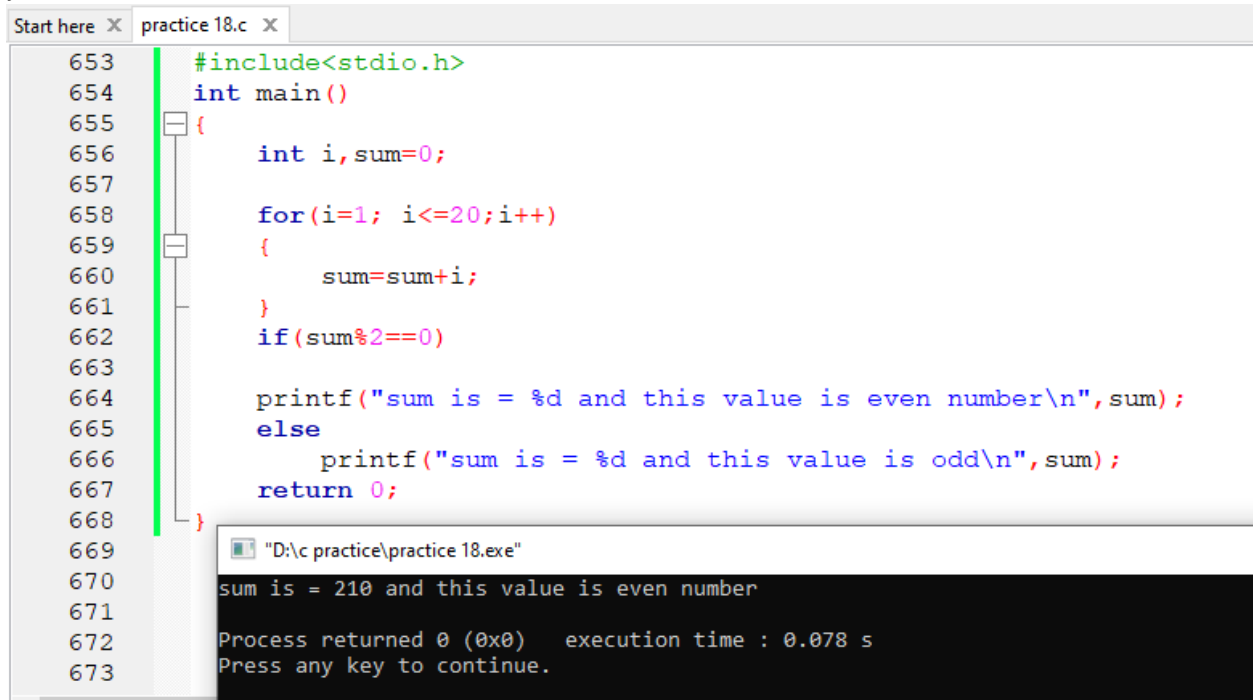
```
        printf("sum is = %d and this value is even number\n",sum);
```

```
    else
```

```
        printf("sum is = %d and this value is odd\n",sum);
```

```
    return 0;
```

```
}
```



The screenshot shows a code editor window with a tab labeled 'practice 18.c'. The code is a C program to calculate the sum of numbers from 1 to 20 and check if it is even or odd. The code is as follows:

```
653 #include<stdio.h>
654 int main()
655 {
656     int i,sum=0;
657
658     for(i=1; i<=20;i++)
659     {
660         sum=sum+i;
661     }
662     if(sum%2==0)
663
664     printf("sum is = %d and this value is even number\n",sum);
665     else
666         printf("sum is = %d and this value is odd\n",sum);
667     return 0;
668 }
```

Below the code editor, there is a console window titled '"D:\c practice\practice 18.exe"'. It displays the output of the program:

```
sum is = 210 and this value is even number
Process returned 0 (0x0)   execution time : 0.078 s
Press any key to continue.
```