

18-05-24

→ main()

```
{ int i, j;
  char x = 'A';
  for (i = 1; i <= 5; i++, printf("\n"), x++)
    for (j = 1; j <= i; j++)
      printf("%c", x);
}
```

O/P:-

```
A
B B
C C C
D D D D
E E E E E
```

→ main()

```
{ int i, j;
  char x = 'A';
  for (i = 1; i <= 5; i++, printf("\n"))
    for (j = 1; j <= i; j++, x++)
      printf("%c", x);
}
```

O/P:-

```
A
B C
D E F
G H I J
K L M N O
```

→ main()

```
{ int i, j;
  char x = 'A';
  for (i = 1; i <= 5; i++, printf("\n"))
    for (x = 'A', j = 1; j <= i; j++, x++)
      printf("%c", x);
}
```

O/P:-

```
A
A B
A B C
A B C D
A B C D E
```

i <= 7

O/P:-

```
A
B C
D E F
G H I J
K L M N O
P Q R S T U
V W X Y Z E I
```

→ int i, j;

for (i = 1; i <= 5; i++, printf("\n"))

for (j = 1; j <= i; j++)

printf("%c", i);

1

2 2

3 3 3

4 4 4 4

5 5 5 5 5



→ main ( )

```

{
    int i, j;
    for (i=1; i<=5; i++, printf("\n"))
    {
        for (j=1; j<=6-i; j++)
            printf(" ");
        for (j=1; j<=i; j++)
            printf("* ");
    }
}

```

O/P:-



~~\*\*\*\*\*~~

↓

m=5

### ⇒ PASCAL TRIANGLE :-

→ main ( )

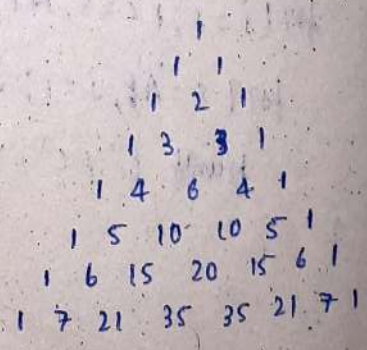
```

{
    int i, j, b, m;
    printf("enter m");
    scanf("%d", &m);
    for (i=0; i<=m; i++, printf("\n"))
    {
        for (j=0; j<=m+1-i; j++)
            printf(" ");
        for (j=0; j<=i; j++)
        {
            if (j==0)
                b=1;
            else
                b = b*(i-j+1)/j;
            printf("%d ", b);
        }
    }
}

```

O/P:-

Enter m: 7





## HOLLOW DIAMOND STAR PATTERN:-

→ main ( )

```
{
    int i, j, m;
    printf("enter m: ");
    scanf("%d", &m);
    // Top half loop
    for (i = 0; i <= m; i++) // iterate rows
    {
        for (j = 0; j <= 2 * m + 1; j++) // iterate columns
        {
            if ((m - i < j) && (j < m + 1 + i)) // spaces are printed within
                printf(" "); // the boundaries of diamond shape.
            else // upper half & middle row of the
                printf("*"); // diamond
        }
        printf("\n");
    }
    // Bottom half loop
    for (i = 0; i <= m; i++) // iterate rows
    {
        for (j = 0; j <= 2 * m + 1; j++) // iterate columns
        {
            if ((i < j) && (j < 2 * m + 1 - i)) // spaces are printed
                printf(" "); // within the boundaries of
            else // diamond shape. Lower half
                printf("*"); // of the diamond
        }
        printf("\n");
    }
}
```

# HOLLOW DIAMOND PATTERN ANY CHARACTER =>

-> main ( )

```
{
    int i, j, m;
    char ch;
    printf ("Enter any character: ");
    scanf ("%c", &ch);
    printf ("enter m: ");
    scanf ("%d", &m);
    for (i = 0; i <= m; i++)
    {
        for (j = 0; j <= 2 * m + 1; j++)
        {
            if ((m - i < j) & (j < m + 1 + i))
                printf (" ");
            else
                printf ("%c", ch);
        }
        printf ("\n");
    }
    for (i = 0; i <= m; i++)
    {
        for (j = 0; j <= 2 * m + 1; j++)
        {
            if ((i < j) & (j < 2 * m + 1 - i))
                printf (" ");
            else
                printf ("%c", ch);
        }
        printf ("\n");
    }
}
```



→ main ( )

```
{ int i, j, m, num = 1;
  printf ("Enter no of rows: ");
  scanf ("%d", &m);
  for (i = 1; i <= m; i++)
  { for (j = 1; j <= i; ++j)
    { printf ("%d", num);
      ++num;
    }
    printf ("\n");
  }
}
```

O/P:-

Enter no of rows: 4

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

For 1-10.0

rows: 14

DIAMOND PATTERN FOR ANY CHARACTER =>

→ main ( )

```
{ int i, j, k;
  char ch;
  printf ("Enter any character: ");
  scanf ("%c", &ch);
  printf ("Enter no of rows: ");
  scanf ("%d", &i);
  for (k = 1; k <= i; k++)
  { for (j = 1; j <= i - k; j++)
    printf (" ");
    for (j = 1; j <= 2 * k - 1; j++)
    printf ("%c", ch);
    printf ("\n");
  }
}
```



```
for(k=1; k<=i-1; k++)
```

```
{
```

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

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

```
for(j=1; j<=2*(i-k)-1; j++)
```

```
printf("%c", ch);
```

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

```
}
```

```
}
```

O/p:-

Enter any character: \$

Enter no of rows: 5

```

      $
    $ $ $
  $ $ $ $ $
$ $ $ $ $ $ $
  $ $ $ $ $ $
    $ $ $ $
      $ $ $
        $
```