**Pattern**

* **Triangle/Right Half Pyramid Pattern :**

1

1 2

1 2 3

1 2 3 4

1 2 3 4 5

Code:

#include<stdio.h>

void main ()

{

int c,r,x=1;

for (r=1;r<=5;r++)

{

for (c=1;c<=r;c++)

{

printf("%d ",c);

}

printf("\n");

}

}

* **Square:**

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

Code:

#include<stdio.h>

void main ()

{

int c,r,x=1;

for (r=1;r<=5;r++)

{

for (c=1;c<=5;c++)

{

printf("%d ",c);

}

printf("\n");

}

}

* **Left Half Pyramid Pattern**

1

12

123

1234

12345

Code:

#include<stdio.h>

void main ()

{

int r,c;

for(r=1;r<=5;r++)

{

for(c=5-1;c>=r;c--)

{

printf(" ");

}

for(c=1;c<=r;c++)

{

printf("%d",c);

}

printf("\n");

}

}

* **Full Pyramid Pattern**

1

1 2

1 2 3

1 2 3 4

1 2 3 4 5

Code:

#include<stdio.h>

void main ()

{

int r,c;

for (r=1;r<=5;r++)

{

for (c=5-1;c>=r;c--)

{

printf(" ");

}

for (c=1;c<=r;c++)

{

printf ("%d ",c);

}

printf("\n");

}

}

* **inverted Right Half Pyramid Pattern**

12345

1234

123

12

1

Code:

#include<stdio.h>

void main ()

{

int r,c;

for(r=5;r>=1;r--)

{

for(c=1;c<=r;c++)

{

printf("%d",c);

}

printf("\n");

}

}

* **Inverted Left Half Pyramid Pattern**

12345

1234

123

12

1

Code:

#include<stdio.h>

void main ()

{

int r,c;

for(r=5;r>=1;r--)

{

for(c=5-1;c>=r;c--)

{

printf(" ");

}

for(c=1;c<=r;c++)

{

printf("%d",c);

}

printf("\n");

}

}

* **Inverted Full Pyramid Pattern**

1 2 3 4 5

1 2 3 4

1 2 3

1 2

1

Code:

#include<stdio.h>

void main ()

{

int r,c;

for (r=5;r>=1;r--)

{

for (c=5-1;c>=r;c--)

{

printf(" ");

}

for (c=1;c<=r;c++)

{

printf ("%d ",c);

}

printf("\n");

}

}

* **Rhombus Pattern**

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

Code:

#include<stdio.h>

void main ()

{

int r,c;

for(r=1;r<=5;r++)

{

for(c=5-1;c>=r;c--)

{

printf(" ");

}

for(c=1;c<=5;c++)

{

printf("%d ",c);

}

printf("\n");

}

}

* **Diamond Pattern**

#include<stdio.h>

void main ()

{

int r,c;

for (r=1;r<=10;r++)

{

for (c=10-1;c>=r;c--)

{

printf(" ");

}

for (c=1;c<=r;c++)

{

printf ("%d ",c);

}

printf("\n");

}

for (r=9;r>=1;r--)

{

for (c=10-1;c>=r;c--)

{

printf(" ");

}

for (c=1;c<=r;c++)

{

printf ("%d ",c);

}

printf("\n");

}

}

* **Hourglass Pattern**

#include<stdio.h>

void main ()

{

int r,c;

for (r=10;r>=2;r--)

{

for (c=10-1;c>=r;c--)

{

printf(" ");

}

for (c=1;c<=r;c++)

{

printf ("%d ",c);

}

printf("\n");

}

for (r=1;r<=10;r++)

{

for (c=10-1;c>=r;c--)

{

printf(" ");

}

for (c=1;c<=r;c++)

{

printf ("%d ",c);

}

printf("\n");

}

}

* **Hollow Square Pattern**

#include<stdio.h>

void main ()

{

int r,c;

for(r=1;r<=10;r++)

{

for(c=1;c<=10;c++)

{

if (r>1 && r<=10-1 && c>1 && c<=10-1)

{

printf(" ");

}

else

{

printf("%d ",c);

}

}

printf("\n");

}

}

* **Hollow Full Pyramid Pattern**

#include<stdio.h>

void main ()

{

int r,c;

for (r=1;r<=10;r++)

{

for (c=10-1;c>=r;c--)

{

printf(" ");

}

for (c=1;c<=r;c++)

{

if (r>1 && r<=10-1 && c>1 && c<r)

{

printf(" ");

}

else

{

printf("%d ",c);

}

}

printf("\n");

}

}

* **Hollow Inverted Full Pyramid Pattern**

#include<stdio.h>

void main ()

{

int r,c;

for (r=10;r>=1;r--)

{

for (c=10-1;c>=r;c--)

{

printf(" ");

}

for (c=1;c<=r;c++)

{

if (r>1 && r<=10-1 && c>1 && c<r)

{

printf(" ");

}

else

{

printf("%d ",c);

}

}

printf("\n");

}

}

* **Hollow Diamond Pattern**

#include<stdio.h>

void main ()

{

int r,c;

for (r=1;r<=10;r++)

{

for (c=10-1;c>=r;c--)

{

printf(" ");

}

for (c=1;c<=r;c++)

{

if (r>1 && r<=10 && c>1 && c<r)

{

printf(" ");

}

else

{

printf("%d ",c);

}

}

printf("\n");

}

for (r=10-1;r>=1;r--)

{

for (c=10-1;c>=r;c--)

{

printf(" ");

}

for (c=1;c<=r;c++)

{

if (r>1 && r<=10 && c>1 && c<r)

{

printf(" ");

}

else

{

printf("%d ",c);

}

}

printf("\n");

}

}

* **Hollow Hourglass Pattern**

#include<stdio.h>

void main ()

{

int r,c;

for (r=10;r>=2;r--)

{

for (c=10-1;c>=r;c--)

{

printf(" ");

}

for (c=1;c<=r;c++)

{

if (r>1 && r<=10-1 && c>1 && c<r)

{

printf(" ");

}

else

{

printf("%d ",c);

}

}

printf("\n");

}

for (r=1;r<=10;r++)

{

for (c=10-1;c>=r;c--)

{

printf(" ");

}

for (c=1;c<=r;c++)

{

if (r>1 && r<=10-1 && c>1 && c<r)

{

printf(" ");

}

else

{

printf("%d ",c);

}

}

printf("\n");

}

}

* **Floyd’s Triangle Pattern**

#include<stdio.h>

void main ()

{

int c,r,x=1;

for (r=1;r<=5;r++)

{

for (c=1;c<=r;c++)

{

printf("%d ",x);

x++;

}

printf("\n");

}

}

* **Pascal’s Triangle Pattern**

#include<stdio.h>

void main ()

{

int c,r;

for (r=1;r<=5;r++)

{

for(c=5-1;c>=r;c--)

{

printf(" ");

}

int x=1;

for (c=1;c<=r;c++)

{

printf("%d ",x);

x=x\*(r-c)/c;

}

printf("\n");

}

}

* **1 to 3 pyramid**

#include<stdio.h>

void main ()

{

int r,c;

for (r=1;r<=15;r++)

{

for (c=14;c>=r;c--)

{

printf(" ");

}

for (c=1;c<=(2\*r-1);c++)

{

printf ("\* ");

}

printf("\n");

}

}

* **Number pattern**

1

1 2 1

1 2 3 2 1

1 2 3 4 3 2 1

1 2 3 4 5 4 3 2 1

Code:

#include<stdio.h>

void main ()

{

int r,c;

for(r=1;r<=5;r++)

{

for(c=1;c<=r;c++)

{

printf("%d ",c);

}

for(c=r-1;c>=1;c--)

{

printf("%d ",c);

}

printf("\n");

}

}

* **Binary pattern**

#include <stdio.h>

int main()

{

int r,c;

for (r=1;r<=4;r++)

{

for (c=1;c<=r;c++)

{

if ((r+c)%2==0)

{

printf("1 ");

}

else

{

printf("0 ");

}

}

printf("\n");

}

return 0;

}

* **Middle print**

x x x x x

x x x x x

x x o x x

x x x x x

x x x x x

Code:

#include <stdio.h>

int main()

{

int r,c;

for (r=1;r<=5;r++)

{

for (c=1;c<=5;c++)

{

if (r==3 && c==3)

{

printf("o ");

}

else

{

printf("x ");

}

}

printf("\n");

}

return 0;

}