Square Star Pattern

#include <stdio.h>

int main()

{

int i, j, num;

printf("Enter a number: ");

scanf("%d",&num);

printf("\n");

for(i = 0; i < num; i++)

{

for(j = 0; j < num; j++)

{

printf("\*");

}

printf("\n");

}

return 0;

}

Enter a number: 5

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

Hollow Pyramid Star Pattern

We challenge you, the programmer to do the hollow pyramid star pattern using C programming

hollow-pyramid-star-pattern.c

#include<stdio.h>

int main()

{

int i, j, k, num;

printf("Enter a number : ");

scanf("%d", &num);

printf("\n");

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

{

for(k = 0; k < num-i; k++)

{

printf(" ");

}

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

{

if(i == num || j == 1 || j == (2\*i-1))

{

printf("\*");

}

else

{

printf(" ");

}

}

printf("\n");

}

return 0;

}

Enter a number: 5

\*

\* \*

\* \*

\* \*

\*\*\*\*\*\*\*\*\*

Inverted Hollow Pyramid Star Pattern Program

#include<stdio.h>

int main()

{

int i, j, k, num;

printf("Enter a number : ");

scanf("%d", &num);

printf("\n");

for(i = num; i >= 1; i--)

{

for(k = i; k < num; k++)

{

printf(" ");

}

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

{

if(i == num || j == 1 || j == (2\*i-1))

{

printf("\*");

}

else

{

printf(" ");

}

}

printf("\n");

}

return 0;

}

Enter a number: 5

\*\*\*\*\*\*\*\*\*

\* \*

\* \*

\* \*

\*

Pyramid Star Pattern

We challenge you, the programmer to do the Pyramid star pattern using C programming

C Programming logic

**Step 1:** Consider every single space as **0**.

**0000**\*

**000**\*\*\*

**00**\*\*\*\*\*

**0**\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*

**Step 2:** From the above star Pattern, we can easily identify, this c program contains 3 for loop.

* for loop represents every new line.
* for loop for displaying 0.
* for loop for displaying \*.

Facts

* C program is not capable of transferring its control from line 2 to line 1.
* Thus the programmer need to complete the line 1 before executing the line 2

C Pyramid Star Pattern - 1

pyramid-star-pattern-1.c

#include<stdio.h>

int main()

{

int i, j, k;

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

{

for(j = 5;j > i;j--)

{

printf("0");

}

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

{

printf("\*");

}

printf("\n");

}

return 0;

}

0000\*

000\*\*\*

00\*\*\*\*\*

0\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*

C Program - Pyramid Star Pattern

Just replace 0 with an empty space.

pyramid-star-pattern.c

#include<stdio.h>

int main()

{

int i, j, k;

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

{

for(j = 5;j > i;j--)

{

printf(" ");

}

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

{

printf("\*");

}

printf("\n");

}

return 0;

}

\*

\*\*\*

\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*

Inverted Pyramid Star Pattern

We challenge you, the programmer to do the inverted pyramid star pattern using C programming

C Programming logic

**step 1:** Consider every single space as **0**.

\*\*\*\*\*\*\*\*\*

**0**\*\*\*\*\*\*\*

**00**\*\*\*\*\*

**000**\*\*\*

**0000**\*

**step 2:** From the above star Pattern, we can easily identify, this C program contains 3 for loop.

* for loop represents every new line.
* for loop for displaying 0.
* for loop for displaying \*.

Facts

* C program is not capable of transferring its control from line 2 to line 1.
* Thus the programmer need to complete the line 1 before executing the line 2

Inverted Pyramid Star Pattern Logically

inverted-pyramid-star.c

#include<stdio.h>

int main()

{

int i, j, k, m = 10;

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

{

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

{

printf("0");

}

for(k = 1;k < m;k++)

{

printf("\*");

}

m = m - 2;

printf("\n");

}

return 0;

}

\*\*\*\*\*\*\*\*\*

0\*\*\*\*\*\*\*

00\*\*\*\*\*

000\*\*\*

0000\*

C Program - Inverted Pyramid Star Pattern

Just replace 0 with an empty space.

inverted-pyramid-star-pattern.c

#include<stdio.h>

int main()

{

int i, j, k, m = 10;

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

{

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

{

printf(" ");

}

for(k = 1;k < m;k++)

{

printf("\*");

}

m = m - 2;

printf("\n");

}

return 0;

}

\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*

\*\*\*

\*

Mirror Half Diamond Star Pattern

We challenge you, the programmer to do the mirror half diamond star pattern using C programming

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

\*\*\*\*

\*\*\*

\*\*

\*

C Programming logic

Lets code a C program to print mirror half diamond star pattern to have some fun

**step 1:** Split this program into 2 parts.

**step 2:** Part one contains.

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

**step 3:** Part two contains.

\*\*\*\*

\*\*\*

\*\*

\*

**step 4:** From the above star pattern, we can easily identify this c program contains 3 for loop.

* for loop represents every new line.
* for loop for displaying 0.
* for loop for displaying \*.

**step 5:** Finally, combine these two star pattern.

Facts

* C program is not capable of transferring its control from line 2 to line 1.
* Thus the programmer need to complete the line 1 before executing the line 2.

Mirror Half Diamond Star Pattern - Part 1

mirror-half-diamond-star-pattern-1

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

mirror-half-diamond-star-pattern-1.c

#include<stdio.h>

int main()

{

int i, j, k;

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

{

for(j = 5;j > i;j--)

{

printf(" ");

}

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

{

printf("\*");

}

printf("\n");

}

return 0;

}

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

Mirror Half Diamond Star Pattern - Part 2

mirror-half-diamond-star-pattern-2

\*\*\*\*

\*\*\*

\*\*

\*

mirror-half-diamond-star-pattern-2.c

#include<stdio.h>

int main()

{

int i, j, k;

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

{

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

{

printf(" ");

}

for(k = 4;k >= i;k--)

{

printf("\*");

}

printf("\n");

}

return 0;

}

\*\*\*\*

\*\*\*

\*\*

\*

Mirror Half Diamond Star Pattern

mirror-half-diamond-star-pattern

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

\*\*\*\*

\*\*\*

\*\*

\*

mirror-half-diamond-star-pattern.c

#include<stdio.h>

int main()

{

int i, j, k;

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

{

for(j = 5;j > i;j--)

{

printf(" ");

}

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

{

printf("\*");

}

printf("\n");

}

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

{

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

{

printf(" ");

}

for(k = 4;k >= i;k--)

{

printf("\*");

}

printf("\n");

}

return 0;

}

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

\*\*\*\*

\*\*\*

\*\*

\*

## C Program - Half Diamond Star Pattern

We challenge you, the programmer to do the half diamond star pattern using C programming

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

\*\*\*\*

\*\*\*

\*\*

\*

## C Programming Logic

**step 1:** Split this program into 2 parts.

**step 2:** Part one contains.

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

**step 3:** Part two contains.

\*\*\*\*

\*\*\*

\*\*

\*

**step 4:** From the above star pattern, we can easily identify, this c program contains 3 for loop.

* for loop represents every new line.
* for loop for displaying 0.
* for loop for displaying \*.

**step 5:** Finally, combine these two star Pattern.

### Facts

* C program is not capable of transferring its control from line 2 to line 1.
* Thus the programmer need to complete the line 1 before executing the line 2

## C Half Diamond Star Pattern - Part 1

Lets code a C program to print half diamond star pattern to have some fun

half-diamond-star-pattern-1

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

half-diamond-star-pattern-1.c

#include<stdio.h>

int main()

{

int i, j;

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

{

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

{

printf("\*");

}

printf("\n");

}

return 0;

}

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

## Half Diamond Star Pattern - Part 2

half-diamond-star-pattern-2

\*\*\*\*

\*\*\*

\*\*

\*

half-diamond-star-pattern-2.c

#include<stdio.h>

int main()

{

int i, j;

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

{

for(j = i;j <= 4;j++)

{

printf("\*");

}

printf("\n");

}

return 0;

}

\*\*\*\*

\*\*\*

\*\*

\*

## Half Diamond Star Pattern - Complete

half-diamond-star-pattern

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

\*\*\*\*

\*\*\*

\*\*

\*

half-diamond-star-pattern.c

#include<stdio.h>

int main()

{

int i, j;

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

{

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

{

printf("\*");

}

printf("\n");

}

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

{

for(j = i;j <= 4;j++)

{

printf("\*");

}

printf("\n");

}

return 0;

}

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

\*\*\*\*

\*\*\*

\*\*

\*

Inverted Right Angle Triangle Star Pattern

We challenge you, the programmer to do the inverted right angle triangle star pattern using C programming

C Programming logic

From the above star Pattern, we can easily identify, this c program contains only 2 for loop.

* for loop represents every new line.
* for loop for diplaying \*.

Facts

* C program is not capable of transferring its control from line 2 to line 1.
* Thus the programmer need to complete the line 1 before executing the line 2.

inverted-right-angle-triangle.c

#include<stdio.h>

int main()

{

int i, j;

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

{

for(j = i;j <= 5;j++)

{

printf("\*");

}

printf("\n");

}

return 0;

}

\*\*\*\*\*

\*\*\*\*

\*\*\*

\*\*

\*

Reverse Mirror Right Angle Triangle Star Pattern

We challenge you, the programmer to do the reverse mirror right angle triangle star pattern using C programming

C Programming logic

1. Consider every single space as **0**.

\*\*\*\*\*

**0**\*\*\*\*

**00**\*\*\*

**000**\*\*

**0000**\*

2. From the above star Pattern, we can easily identify, this c program contains 3 for loop.

* for loop represents every new line.
* for loop for displaying 0.
* for loop for displaying \*.

Facts about a Pattern

* C program is not capable of transferring its control from line 2 to line 1.
* Thus the programmer need to complete the line 1 before executing the line 2.

Reverse Mirror Right Angle Triangle Star Pattern - Part 1

reverse-mirror-right-angle-triangle-1.c

#include<stdio.h>

int main()

{

int i, j, k;

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

{

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

{

printf("0");

}

for(k = 5;k >= i;k--)

{

printf("\*");

}

printf("\n");

}

return 0;

}

\*\*\*\*\*

0\*\*\*\*

00\*\*\*

000\*\*

0000\*

Reverse Mirror Right Angle Triangle Star Pattern

Just replace 0 with an empty space

reverse-mirror-right-angle-triangle.c

#include<stdio.h>

int main()

{

int i, j, k;

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

{

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

{

printf(" ");

}

for(k = 5;k >= i;k--)

{

printf("\*");

}

printf("\n");

}

return 0;

}

\*\*\*\*\*

\*\*\*\*

\*\*\*

\*\*

\*

Diamond Star Pattern

We challenge you, the programmer to do the diamond star pattern using c programming

C Programming logic

1) Split this program into 2 parts.

2) part one contains.

\*

\*\*\*

\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*

3) part two contains.

\*\*\*\*\*\*\*

\*\*\*\*\*

\*\*\*

\*

Diamond Star Pattern Program - Part 1

Consider every single space in part 1 as **0**.

diamond-star-pattern-1

**0000**\*

**000**\*\*\*

**00**\*\*\*\*\*

**0**\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*

diamond-star-pattern-1.c

#include<stdio.h>

int main()

{

int i, j, k;

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

{

for(j = 5;j > i;j--)

{

printf("0");

}

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

{

printf("\*");

}

printf("\n");

}

return 0;

}

0000\*

000\*\*\*

00\*\*\*\*\*

0\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*

Diamond Star Pattern - Part 2

diamond-star-pattern-2

0\*\*\*\*\*\*\*

00\*\*\*\*\*

000\*\*\*

0000\*

diamond-star-pattern-2.c

#include<stdio.h>

int main()

{

int i, j, k, m = 8;

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

{

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

{

printf("0");

}

for(k = i;k < m;k++)

{

printf("\*");

}

m-=1;

printf("\n");

}

return 0;

}

0\*\*\*\*\*\*\*

00\*\*\*\*\*

000\*\*\*

0000\*

Diamond Star Pattern - Combined

diamond-star-pattern-comb

0000\*

000\*\*\*

00\*\*\*\*\*

0\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*

0\*\*\*\*\*\*\*

00\*\*\*\*\*

000\*\*\*

0000\*

diamond-star-pattern-comb.c

#include<stdio.h>

int main()

{

int i, j, k, m = 8;

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

{

for(j = 5;j > i;j--)

{

printf("0");

}

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

{

printf("\*");

}

printf("\n");

}

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

{

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

{

printf("0");

}

for(k = i;k < m;k++)

{

printf("\*");

}

m-=1;

printf("\n");

}

return 0;

}

0000\*

000\*\*\*

00\*\*\*\*\*

0\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*

0\*\*\*\*\*\*\*

00\*\*\*\*\*

000\*\*\*

0000\*

Diamond Star Pattern - Complete

diamond-star-pattern

\*

\*\*\*

\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*

\*\*\*

\*

diamond-star-pattern.c

#include<stdio.h>

int main()

{

int i, j, k, m = 8;

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

{

for(j = 5;j > i;j--)

{

printf(" ");

}

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

{

printf("\*");

}

printf("\n");

}

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

{

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

{

printf(" ");

}

for(k = i;k < m;k++)

{

printf("\*");

}

m-=1;

printf("\n");

}

return 0;

}

\*

\*\*\*

\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*

\*\*\*

\*