1. **Write a program to print solid and hollow rectangle star pattern**

**Algorithm:**

step 1: START.  
step 2: Initialize variables.  
step 3: Check **for** Condition. If its **true**, **goto** step 4 otherwise **goto** step 7.  
step 4: Check **for** Condition.If its **true**,**goto** step 5 otherwise **goto** step 6.  
step 5: Print **"\*" and repeat step 4**  
step 6: Goto next line and **repeat step 3**  
step 7: STOP.

**Program:**

**package** \_6.Patterns\_Programs;

**import** java.util.Scanner;

**public** **class** Solid\_Rectangle {

**static** **void** solid\_rectangle(**int** n, **int** m) {

**int** i, j;

**for** (i = 1; i <= n; i++) {

**for** (j = 1; j <= m; j++) {

System.***out***.print("\*");

}

System.***out***.print("\n");

}

}

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

**int** rows, columns;

Scanner sc = **new** Scanner(System.***in***);

System.***out***.print("\nEnter the number of rows :" );

rows = sc.nextInt();

System.***out***.print("\nEnter the number of columns : ");

columns = sc.nextInt();

System.***out***.print("\n");

*solid\_rectangle*(rows, columns);

}

}

**Expected O/P:**

Enter the number of rows :5

Enter the number of columns : 6

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

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

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

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

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

1. **Write a program using Hollow rectangular star pattern?**

**Algorithm:**

step 1: START

step 2: Initialize variables.

step 3: Check for Condition.If its true,goto step 4 otherwise step 8

step 4: Check for Condition.If its true,then goto step 5 otherwise goto step 7

step 5: Check if condition.If any condition satisfies,print "\*" and repeat step 4 otherwise goto step 6.

step 6: Print space in same line and repeat step 4

step 7: Goto next line and repeat step 3

step 8: STOP..

**Program:**

**package** \_6.Patterns\_Programs;

**import** java.util.Scanner;

**public** **class** Hollow\_Rectangle {

**static** **void** hollow\_rectangle(**int** n, **int** m) {

**int** i, j;

**for** (i = 1; i <= n; i++) {

**for** (j = 1; j <= m; j++) {

**if** (i == 1 || i == n || j == 1 || j == m)

System.***out***.print("\*");

**else**

System.***out***.print(" ");

}

System.***out***.print("\n");

}

}

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

**int** rows, columns;

Scanner sc = **new** Scanner(System.***in***);

System.***out***.print("\nEnter the number of rows : ");

rows = sc.nextInt();

System.***out***.print("\nEnter the number of columns : ");

columns = sc.nextInt();

System.***out***.print("\n");

*hollow\_rectangle*(rows, columns);

}

}

**Expected O/P:**

Enter the number of rows : 3

Enter the number of columns : 4

\*\*\*\*

\* \*

\*\*\*\*

1. **Program to print pyramid pattern using stars or Pyramid star patterns are discussed here. For any given number n, print various pyramid star patterns.**

**Half Pyramid Star Pattern:**

**Algorithm:**

step 1: START.  
step 2: Initialize variables.  
step 3: Check **for** Condition.If its **true**,**goto** step 4 otherwise **goto** step 7.  
step 4: Check **for** Condition.If it’s **true**,**goto** step 5 otherwise **goto** step 6.  
step 5: Print **"\*" and repeat step 4**  
step 6: Goto next line and **repeat step 3**.  
step 7: STOP.

**Program:**

**package** \_6.Patterns\_Programs;

**import** java.util.Scanner;

**public** **class** HalfPyaramid {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

**int** n, i, j;

Scanner sc = **new** Scanner(System.***in***);

n = sc.nextInt();

**for** (i = 0; i < n; i++) {

**for** (j = 0; j <= i; j++) {

System.***out***.print("\*");

}

System.***out***.print("\n");

}

}

}

**Expected O/P:**

4

\*

\*\*

\*\*\*

\*\*\*\*

1. **Write a program for Inverted Half pyramid?**

**Algorithm:**

step 1: START.

step 2: Initialize variables.  
step 3: Check **for** Condition.If its **true**,**goto** step 4 otherwise **goto** step 7.  
step 4: Check **for** Condition.If it’s **true**,**goto** step 5 otherwise **goto** step 6.  
step 5: Print **"\*" and repeat** step 4  
step 6: Goto next line and **repeat** step 3  
step 7: STOP.

**Program:**

**package** \_6.Patterns\_Programs;

**import** java.util.Scanner;

**public** **class** InvertedHalfPyramid {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

**int** n, i, j, k = 0;

Scanner sc = **new** Scanner(System.***in***);

n = sc.nextInt();

**for** (i = n; i >= 1; i--) {

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

System.***out***.print("\*");

}

System.***out***.print("\n");

}

}

}

**Expected O/P:**

5

\*\*\*\*\*

\*\*\*\*

\*\*\*

\*\*

\*

1. **Write a program for Full Pyramid Star Pattern?**

**Algorithm:**

step 1: START  
step 2: Initialize variables.  
step 3: Check **for** condition.If it’s **true**,**goto** step 4 otherwise **goto** step 10.   
step 4: Check **for** condition.If it’s **true**,**goto** step 5 otherwise **goto** step 9.  
step 5:Print space in same line and **repeat** step 4.  
step 6: Check **while** condition.If it’s **true**,**goto** step 7 otherwise **goto** step 9.  
step 7: Print **"\*"** and **goto** step 8.  
step 8: Increment the value and **repeat** step 6.  
step 9: Goto next line and **repeat** step 3.  
step 10: STOP.

**Program:**

**package** \_6.Patterns\_Programs;

**import** java.util.Scanner;

**public** **class** FullPyaramid {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

**int** n, i, j, k = 0;

Scanner sc = **new** Scanner(System.***in***);

n = sc.nextInt();

**for** (i = 1; i <= n; ++i, k = 0) {

**for** (j = 1; j <= n - i; ++j) {

System.***out***.print(" ");

}

**while** (k != 2 \* i - 1) {

System.***out***.print("\*");

++k;

}

System.***out***.print("\n");

}

}

}

**Expected O/P:**

5

\*

\*\*\*

\*\*\*\*\*

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

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

1. **Write a program using Inverted Full pyramid?**

**Algorithm:**

step 1: START.

step 2: Initialize variables.

step 3: Check for loop condition.If it’s true,goto step 4 otherwise goto step 11.

step 4: Check for loop condition.If it’s true,goto step 5 otherwise goto step 6.

step 5: Print space and repeat step 4

step 6: Check for loop condition.If it’s true,goto step 7 otherwise goto step 8.

step 7: Print "\*" and repeat step 6

step 8: Check for loop condition.If it’s true,goto step 9 otherwise goto step 10.

step 9: Print "\*" and repeat step 8

step 10: Goto next line and repeat step 3

step 11: STOP.

**Program:**

**package** \_6.Patterns\_Programs;

**import** java.util.Scanner;

**public** **class** Inverted\_Full {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

**int** n, i, j, k = 0;

Scanner sc = **new** Scanner(System.***in***);

n = sc.nextInt();

**for** (i = n; i >= 1; --i) {

**for** (j = 0; j < n - i; ++j) {

System.***out***.print(" ");

}

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

System.***out***.print("\*");

}

**for** (j = 0; j < i - 1; ++j) {

System.***out***.print("\*");

}

System.***out***.print("\n");

}

}

}

**Expected O/P:**

5

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

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

\*\*\*\*\*

\*\*\*

\*

1. **Write a program for Hollow Full Pyramid Star Pattern**

**Algorithm:**

step 1: START

step 2: Initialize variables

step 3: Check for loop condition.If it's true,goto step 4 otherwise goto step 12

step 4: Check for loop condition.If it's true,goto step 5 else goto step 6

step 5: Print space and repeat step 4

step 6: Check the condition of while loop.If it's true,goto step 7 otherwise goto step

step 7: Check the condition of if statement. If it's true,goto step 8 else goto step 9.

step 8: Print "\*" and goto step 10

step 9: Print space and goto step 10.

step 10: Increment the value and repeat step 6

step 11: Assign k value to 0 and print next row.Repeat step 3.

step 12: Check for loop condition and if it's true,goto step 13 otherwise goto step 14.

step 13: Print "\*" and repeat step 12

step 14: STOP.

**Program:**

**package** \_6.Patterns\_Programs;

**import** java.util.Scanner;

**public** **class** HollowFull {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

**int** n;

Scanner sc = **new** Scanner(System.***in***);

n = sc.nextInt();

**int** i, j, k = 0;

**for** (i = 1; i <= n; i++) {

**for** (j = i; j < n; j++) {

System.***out***.print(" ");

}

**while** (k != (2 \* i - 1)) {

**if** (k == 0 || k == 2 \* i - 2) {

System.***out***.print("\*");

} **else** {

System.***out***.print(" ");

}

k++;

}

k = 0;

System.***out***.print("\n"); // print next row

}

**for** (i = 0; i < 2 \* n - 1; i++) {

System.***out***.print("\*");

}

}

}

**Expected O/P:**

5

\*

\* \*

\* \*

\* \*

\* \*

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

1. **Write a program for inverted Hollow Half pyramid?**

**Algorithm:**

step 1: START

step 2: Initialize variables

step 3: Check for loop condition.If it's true,goto step 4 otherwise goto step 12

step 4: Check for loop condition.If it's true,goto step 5 else goto step 6

step 5: Print space and repeat step 4

step 6: Check the condition of while loop.If it's true,goto step 7 otherwise goto step

step 7: Check the condition of if statement. If it's true,goto step 8 else goto step 9.

step 8: Print "\*" and goto step 10

step 9: Print space and goto step 10.

step 10: Increment the value and repeat step 6

step 11: Assign k value to 0 and print next row.Repeat step 3.

step 12: Check for loop condition and if it's true,goto step 13 otherwise goto step 14.

step 13: Print "\*" and repeat step 12

step 14: STOP.

**Program:**

**package** \_6.Patterns\_Programs;

**import** java.util.Scanner;

**public** **class** Inverted\_Hollow\_Half {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

**int** n;

Scanner sc = **new** Scanner(System.***in***);

n = sc.nextInt();

**int** i, j, k = 0;

**for** (i = 0; i < n; i++) {

System.***out***.print("\*");

}

**for** (i = n; i >= 1; --i) {

**for** (j = 0; j < n - i; ++j)

**while** (k != (2 \* i - 1)) {

**if** (k == 0 || k == 2 \* i - 2)

System.***out***.print("\*");

**else**

System.***out***.print(" ");

k++;

}

k = 0;

System.***out***.print("\n"); // print next row

}

}

}

**Expected O/P:**

6

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

\* \*

\* \*

\* \*

\* \*

\*

1. **Write a program for Inverted Hollow Full Pyramid?**

**Algorithm:**

step 1: START

step 2: Initialize variables

step 3: Check for loop condition.If it's true,goto step 4 otherwise goto step 11.

step 4: Check for loop condition.If it's true,goto step 5 otherwise goto step 6

step 5: Print "\*" and repeat step 4

step 6: Check for loop condition.If it's true,goto step 7 otherwise goto step

step 7: Check if statement condition it's true,goto step 8 otherwise goto step 9

step 8: Print "\*" and repeat step 6

step 9: Print space and repeat step 6

step 10: Print next row and repeat step 3

step 11: STOP

**Program:**

**package** \_6.Patterns\_Programs;

**import** java.util.Scanner;

**public** **class** Inverted\_Hollow\_Full {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

**int** n;

Scanner sc = **new** Scanner(System.***in***);

n = sc.nextInt();

**int** i, j;

**for** (i = 1; i <= n; i++) {

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

System.***out***.print(" ");

}

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

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

System.***out***.print("\*");

} **else** {

System.***out***.print(" ");

}

}

System.***out***.print("\n");

}

}

}

**Expected O/P:**

5

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

\* \*

\* \*

\* \*

\*

1. **Write a program for Half pyramid pattern using numbers**

**Algorithm:**

step 1: START  
step 2: Initialize variables  
step 3: Check **for** loop condition.If it**'s true,goto step 4 otherwise goto step 7**step 4: Check **for** loop condition.If it**'s true,goto step 5 otherwise goto step 6**step 5: Print j value and **repeat** step 4  
step 6: Print on next line and **repeat** step 3  
step 7: STOP.

**Program:**

**package** \_6.Patterns\_Programs;

**import** java.util.Scanner;

**public** **class** Half\_Numbers {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

**int** n;

Scanner sc = **new** Scanner(System.***in***);

n = sc.nextInt();

**int** i, j;

**for** (i = 1; i <= n; i++) {

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

System.***out***.print(j + " ");

}

System.***out***.println();

}

}

}

**Expected O/P:**

6

1

1 2

1 2 3

1 2 3 4

1 2 3 4 5

1 2 3 4 5 6

1. **Write a program using Inverted half pyramid pattern using numbers?**

**Algorithm:**

step 1: START  
step 2: Initialize variables  
step 3: Check **for** loop condition.If it**'s true,goto step 4 otherwise goto step 7**step 4: Check **for** loop condition.If it**'s true,goto step 5 otherwise goto step 6**step 5: Print j value and **repeat** step 4  
step 6: Print on next line and **repeat** step 3  
step 7: STOP.

**Program:**

**package** \_6.Patterns\_Programs;

**import** java.util.Scanner;

**public** **class** Inverted\_Half\_number {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

**int** n;

Scanner sc = **new** Scanner(System.***in***);

n = sc.nextInt();

**int** i,j;

**for**(i = 1; i<= n; i++)

{

**for**(j = i; j <= n; j++)

{

System.***out***.print(j + " ");

}

System.***out***.println();

}

}

}

**Expected O/P:**

6

1 2 3 4 5 6

2 3 4 5 6

3 4 5 6

4 5 6

5 6

6

1. **Write a program for Full pyramid pattern using numbers?**

**Algorithm:**

step 1: START

step 2: Initialize variables.

step 3: Check for loop condition.If it's true,goto step 4 otherwise goto step 12

step 4: Check for loop condition.If it's true,goto step 5 otherwise goto step 6

step 5: Print space and increment count value. Repeat step 4.

step 6: Check while loop condition.If it's true, goto step 7 otherwise goto step 11.

step 7: Check if statement condition.If it's true,goto step 8 otherwise goto step 9

step 8: Print i+k value and increment count variable. Goto step 10

step 9: Increment count1 variable and print i+k-2\*count1 value. Goto step 10.

step 10: Increment k value and repeat step 6

step 11: Assign count,count1,k values to 0 and Print in next line and repeat step 3

step 12: STOP.

**Program:**

**package** \_6.Patterns\_Programs;

**import** java.util.Scanner;

**public** **class** Full\_number {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

**int** i, j, n, count = 0, count1 = 0, k = 0;

Scanner sc = **new** Scanner(System.***in***);

n = sc.nextInt();

**for** (i = 1; i <= n; ++i) {

**for** (j = 1; j <= n - i; ++j) {

System.***out***.print(" ");

++count;

}

**while** (k != 2 \* i - 1) {

**if** (count <= n - 1) {

System.***out***.print(i + k + " ");

++count;

} **else** {

++count1;

System.***out***.print((i + k - 2 \* count1) + " ");

}

++k;

}

count1 = count = k = 0;

System.***out***.println();

}

}

}

**Expected O/P:**

6

1

2 3 2

3 4 5 4 3

4 5 6 7 6 5 4

5 6 7 8 9 8 7 6 5

6 7 8 9 10 11 10 9 8 7 6

1. **Write a program for Hollow half pyramid pattern using numbers?**

**Algorithm:**

step 1: START  
step 2: Initialize variables  
step 3: Check **for** loop condition.If it**'s true,goto step 4 otherwise goto step 9**step 4: Check **for** loop condition.If it**'s true,goto step 5 otherwise goto step 8**step 5: Check **if** statement condition.If it**'s true,goto step 6 otherwise goto step 7**step 6: Print j value and **repeat** step 4  
step 7: Print space and **repeat** step 4  
step 8: Print in next line and repeatstep 3  
step 9: STOP.

**Program:**

**package** \_6.Patterns\_Programs;

**import** java.util.Scanner;

**public** **class** Hollow\_Half\_Number {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

**int** n, i, j;

Scanner sc = **new** Scanner(System.***in***);

n = sc.nextInt();

**for** (i = 1; i <= n; i++) {

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

**if** (j == 1 || j == i || i == n)

System.***out***.print(j);

**else**

System.***out***.print(" ");

}

System.***out***.println();

}

}

}

**Expected O/P:**

5

1

12

1 3

1 4

12345

1. **Write a program for Hollow Inverted Half Pyramid Pattern using numbers?**

**Algorithm:**

step 1: START  
step 2: Initialize variables  
step 3: Check **for** loop condition.If it**'s true,goto step 4 otherwise goto step 9**step 4: Check **for** loop condition.If it**'s true,goto step 5 otherwise goto step 8**step 5: Check **if** statement condition.If it**'s true,goto step 6 otherwise goto step 7**step 6: Print j value and **repeat** step 4  
step 7: Print space and **repeat** step 4  
step 8: Print in next line and **repeat** step 3  
step 9: STOP.

**Program**:

**package** \_6.Patterns\_Programs;

**import** java.util.Scanner;

**public** **class** HollowInvertedHalfPyramid {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

**int** n, i, j;

Scanner sc = **new** Scanner(System.***in***);

n = sc.nextInt();

**for** (i = 1; i <= n; i++) {

**for** (j = i; j <= n; j++) {

**if** (i == 1 || j == n || j == i)

System.***out***.print(j);

**else**

System.***out***.print(" ");

}

System.***out***.println();

}

}

}

**Expected O/P:**

5

12345

2 5

3 5

45

5

1. **Write a program for Hollow Full Pyramid Pattern using numbers?**

**Algorithm:**

step 1: START

step 2: Initialize variables

step 3: Check for loop condition.If it's true,goto step 4 otherwise goto step 15

step 4: Check for loop condition.If it's true,goto step 5 otherwise goto step 6

step 5: Print space and repeat step 4

step 6: Check for loop condition.If it's true,goto step 7 otherwise goto step 10

step 7: Check if statement condition.If it's true,goto step 8 otherwise goto step 9

step 8: Print j values and repeat step 6.

step 9: Print space and repeat step 6.

step 10: Check for loop condition.If it's true,goto step 11 otherwise goto step 14

step 11: Check if statement condition.If it's true,goto step 12 otherwise goto step 13

step 12: Print j+1 value and repeat step 10

step 13: Print space and repeat step 10

step 14: Print on next line and repeat step 3

step 15: STOP.

**Program:**

**package** \_6.Patterns\_Programs;

**import** java.util.Scanner;

**public** **class** HollowFullPyramidPattern\_numbers {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

**int** n, i, j;

Scanner sc = **new** Scanner(System.***in***);

n = sc.nextInt();

**for** (i = 1; i <= n; i++) {

**for** (j = i; j < n; j++) {

System.***out***.print(" ");

}

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

**if** (j == 1 || i == n) {

System.***out***.print(j + " ");

} **else** {

System.***out***.print(" ");

}

}

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

**if** (j == i - 1 && j < n - 1) {

System.***out***.print(j + 1);

} **else** {

System.***out***.print(" ");

}

}

System.***out***.print("\n");

}

}

}

**Expected O/P:**

5

1

1 2

1 3

1 4

1 2 3 4 5

1. **Write a program for palindrome half pyramid pattern using numbers**

**Algorithm:**

step 1: START  
step 2: Initialize variables  
step 3: Check **for** loop condition.If it**'s true,goto step 4 otherwise goto step 9**step 4: Check **for** loop condition.If it**'s true,goto step 5 otherwise goto step 6**step 5: Print k value and **repeat** step 4  
step 6: Check **for** loop condition.If it**'s true,goto step 7 otherwise goto step 8**step 7: Print l values and **repeat** step 6  
step 8: Print on next line and **repeat** step 3  
step 9: STOP

**Program:**

**package** \_6.Patterns\_Programs;

**import** java.util.Scanner;

**public** **class** Plindromehalfpyramid {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

**int** n, k, l, i;

Scanner sc = **new** Scanner(System.***in***);

System.***out***.print("\nEnter the number of rows : ");

n = sc.nextInt();

System.***out***.print("\n");

**for**(i = 1; i <= n; i++)

{

**for**(k = 1; k <= i; k++)

{

System.***out***.print(k + " ");

}

**for**(l = i-1; l >= 1; l--)

{

System.***out***.print(l + " ");

}

System.***out***.print("\n");

}

}

}

**Expected I/P:**

Enter the number of rows : 4

**Expected O/P:**

1

1 2 1

1 2 3 2 1

1 2 3 4 3 2 1

1. **Write a program for palindrome half pyramid pattern using alphabets?**

**Algorithm:**

step 1: START  
step 2: Initialize variables  
step 3: Check **for** loop condition.If it**'s true,goto step 4 otherwise goto step 9**step 4: Check **for** loop condition.If it**'s true,goto step 5 otherwise goto step 6**step 5: Print character with value k+65-1 and **repeat** step 4  
step 6: Check **for** loop condition.If it**'s true,goto step 7 otherwise goto step 8**step 7: Print character with value l+65-1 and **repeat** step 4  
step 8: Print on next line and **repeat** step 3  
step 9: STOP

**Program:**

**package** \_6.Patterns\_Programs;

**import** java.util.Scanner;

**public** **class** Palindromehalfpyramid\_Number {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

**int** n, k, l, i;

Scanner sc = **new** Scanner(System.***in***);

System.***out***.print("\nEnter the number of rows : ");

n = sc.nextInt();

System.***out***.print("\n");

**for**(i = 1; i <= n; i++)

{

**for**(k = 1; k <= i; k++)

{

System.***out***.print((**char**)(k+65-1) + " ");

}

**for**(l = i-1; l >= 1; l--)

{

System.***out***.print((**char**)(l+65-1) + " ");

}

System.***out***.print("\n");

}

}

}

**Expected I/P:**

Enter the number of rows : 5

**Expected O/P:**

A

A B A

A B C B A

A B C D C B A

A B C D E D C B A

1. **Write a program for palindrome pyramid pattern using numbers**

**Algorithm:**

step 1: START

step 2: Initialize variables  
step 3: Check **for** loop condition.If it**'s true,goto step 4 otherwise goto step 11.**step 4: Check **for** loop condition.If it**'s true,goto step 5 otherwise goto step 6**step 5: Print space and **repeat** step 4  
step 6: Check **for** loop condition.If it**'s true,goto step 7 otherwise goto step**step 7: Check **if** statement condition.If **true**,**goto** step 8 otherwise **goto** step 9  
step 8: Print j value and **repeat** step 6  
step 9: Print k value and **repeat** step 6  
step 10: Print in next value and print space. Repeat step 3  
step 11: STOP

**Program:**

**package** \_6.Patterns\_Programs;

**import** java.util.Scanner;

**public** **class** Palindrom\_Pyramid {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

**int** n, i, j, k;

Scanner sc = **new** Scanner(System.***in***);

n = sc.nextInt();

System.***out***.print(" ");

**for** (i = 1; i <= n; i++) {

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

System.***out***.print(" ");

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

**if** (j <= k)

System.***out***.print(j);

**else**

System.***out***.print(k);

}

System.***out***.print("\n");

System.***out***.print(" ");

}

}

}

**Expected I/p:**

4

**Expected O/P:**

1

121

12321

1234321

1. **Write a program for Palindrome pyramid pattern using numbers and stars**

**Algorithm:**

step 1: START

step 2: Initialize variables

step 3: Check for loop condition.If it's true,goto step 4 otherwise goto step 16

step 4: Check for loop condition.If it's true,goto step 5 otherwise goto step 7

step 5: Check if statement condition.If it's true,goto step 6 otherwise goto step 7

step 6: Print "\*" and repeat step 4

step 7: Increment num value and goto step 8

step 8: Check for loop condition.If it's true,goto step 9 otherwise goto step 12

step 9: Print num and goto step 10.

step 10: Check if statement condition.If it's true,goto step 11 otherwise goto step 12

step 11: Print "\*" and increment count value.Repeat step 8

step 12: Check for loop condition.If it's true,goto step 13 otherwise goto step 16

step 13: Check if statement condition.If it's true,goto step 14 otherwise goto step 15

step 14: Print "\*" and repeat step 12

step 15: Print in next line and decrement space value.Assign count value to 1 and goto step 16

step 16: STOP.

**Program:**

**package** \_6.Patterns\_Programs;

**import** java.util.Scanner;

**public** **class** PalindromepyramidNumbers {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

Scanner sc = **new** Scanner(System.***in***);

**int** i, j, space, count = 1, num = 0, star = 8;

**int** n = sc.nextInt();

space = n;

**for** (i = 1; i <= n; i++)

{

**for** (j = 1; j <= star; j++)

**if**(i + j <= star + 1)

System.***out***.print("\*");

num++;

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

{

System.***out***.print(num);

**if** (i > 1 && count < i)

{

System.***out***.print("\*");

count++;

}

}

**for** (j = 1; j <= star; j++)

**if**(i + n <= j + n)

System.***out***.print("\*");

System.***out***.println();

space--;

count = 1;

}

}

}

**Expected I/P:**

5

**Expected O/P:**

\*\*\*\*\*\*\*\*1\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*2\*2\*\*\*\*\*\*\*

\*\*\*\*\*\*3\*3\*3\*\*\*\*\*\*

\*\*\*\*\*4\*4\*4\*4\*\*\*\*\*

\*\*\*\*5\*5\*5\*5\*5\*\*\*\*

1. **Write a program for Solid Diamond pattern printing using stars?**

**Algorithm:**

step 1: START

step 2: Initialize variables

step 3: Check for loop condition.If it's true,goto step 4 otherwise goto step 10

step 4: Check for loop condition.If it's true,goto step 5 else goto step 6

step 5: Print space and repeat step 4.

step 6: Decrement space value and goto step 7

step 7: Check for loop condition.If it's true,goto step 8 else goto step 9

step 8: Print "\*" and repeat step 7

step 9: Print on next line and repeat step 3

step 10: Assign space value to 1 and goto step 11

step 11: Check for loop condition.If it's true,goto step 12 otherwise goto step 18

step 12: Check for loop condition.If it's true,goto step 13 otherwise goto step 13

step 13: Print space and repeat step 12

step 14: Increment space value and goto step 15

step 15: Check for loop condition.If it's true,goto step 16 else goto step 17

step 16: Print "\*" and repeat step 15

step 17: Print on next line and repeat step 11

step 18: STOP

**Program:**

**package** \_6.Patterns\_Programs;

**import** java.util.Scanner;

**public** **class** SolidDiamond {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

**int** n, c, k, space = 1;

System.***out***.print("\nEnter the number of rows : ");

Scanner sc = **new** Scanner(System.***in***);

n = sc.nextInt();

System.***out***.println();

space = n - 1;

**for** (k = 1; k <= n; k++)

{

**for** (c = 1; c <= space; c++)

System.***out***.print(" ");

space--;

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

System.***out***.print("\*");

System.***out***.print("\n");

}

space = 1;

**for** (k = 1; k <= n - 1; k++)

{

**for** (c = 1; c <= space; c++)

System.***out***.print(" ");

space++;

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

System.***out***.print("\*");

System.***out***.print("\n");

}

}

}

**Expected I/P:**

Enter the number of rows: 4

**Expected O/P:**

\*

\*\*\*

\*\*\*\*\*

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

\*\*\*\*\*

\*\*\*

\*

1. **Write a program for Hollow diamond pattern printing using stars?**

**Algorithm:**

step 1: START

step 2: Initialize values

step 3: Check for loop condition.If true,goto step 4 else goto step 12

step 4: Check for loop condition.If true,goto step 5 else goto step 6

step 5: Print space and repeat step 4

step 6: Check while loop condition.If true,goto step 7 else goto step 11

step 7: Check if statement condition.If true,goto step 8 else goto step 9

step 8: Print "\*" and goto step 10

step 9: Print space and goto step 10

step 10: Increment k value and repeat step 6

step 11: Assign k value to 0 and print in next line.Repeat step 3.

step 12: Decrement n value and goto step 13

step 13: Check for loop condition.If true,goto step 14 else goto step 23

step 14: Check for loop condition.If true,goto step 15 else goto step 16

step 15: Print space and repeat step 14

step 16: Assign k value to 0 and goto step 17

step 17: Check while loop condition.If true,goto step 18 else goto step 22

step 18: Check if statement condition.If true,goto step 19 else goto step 20

step 19: Print "\*" and goto step 21

step 20: Print space and goto step 21

step 21: Increment k value and repeat step 17

step 22: Print on next line and repeat step 13

step 23: STOP.

**Program:**

**package** \_6.Patterns\_Programs;

**import** java.util.Scanner;

**public** **class** Hollowdiamond {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

**int** i, j, space, k = 0, n;

System.***out***.print("\nEnter the number of rows : ");

Scanner sc = **new** Scanner(System.***in***);

n = sc.nextInt();

System.***out***.println();

**for** (i = 1; i <= n; i++) {

**for** (j = 1; j <= n - i; j++) {

System.***out***.print(" ");

}

**while** (k != (2 \* i - 1)) {

**if** ((k == 0) || (k == 2 \* i - 2))

System.***out***.print("\*");

**else**

System.***out***.print(" ");

k++;

}

k = 0;

System.***out***.print("\n");

}

n--;

**for** (i = n; i >= 1; i--) {

**for** (j = 0; j <= n - i; j++) {

System.***out***.print(" ");

}

k = 0;

**while** (k != (2 \* i - 1)) {

**if** ((k == 0) || (k == 2 \* i - 2))

System.***out***.print("\*");

**else**

System.***out***.print(" ");

k++;

}

System.***out***.print("\n");

}

}

}

**Expected I/P:**

Enter the number of rows : 4

**Expected O/P:**

\*

\* \*

\* \*

\* \*

\* \*

\* \*

\*

1. **Write a program for Solid Half Diamond pattern printing using stars?**

**Algorithm:**

step 1: START

step 2: Initialize variables

step 3: Check for loop condition.If it's true, goto step 4 else goto step 9

step 4: Check for loop condition. If true,goto step 5 else goto step 6

step 5: Print space and repeat step 4

step 6: Check for loop condition.If true, goto step 7 else goto step 8

step 7: Print "\*" and repeat step 6

step 8: Print in next line and repeat step 3

step 9: Check for loop condition.If true, goto step 10 else goto step 15

step 10: Check for loop condition. If true,goto step 11 else goto step 12

step 11: Print space and repeat step 10

step 12: Check for loop condition. If true,goto step 13 else goto step 14

step 13: Print space and repeat step 12

step 14: Print on next line and repeat step 9.

step 15: STOP.

**Program:**

**package** \_6.Patterns\_Programs;

**import** java.util.Scanner;

**public** **class** SolidHalfDiamond {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

Scanner sc = **new** Scanner(System.***in***);

System.***out***.println("Enter N : ");

**int** n = sc.nextInt();

**for** (**int** i = 1; i <= n; i++) {

**for** (**int** j = 1; j <= n - i; j++) {

System.***out***.print(" ");

}

**for** (**int** j = 1; j <= i; j++) {

System.***out***.print("\*");

}

System.***out***.println();

}

**for** (**int** i = n - 1; i > 0; i--) {

**for** (**int** j = 1; j <= n - i; j++) {

System.***out***.print(" ");

}

**for** (**int** j = 1; j <= i; j++) {

System.***out***.print("\*");

}

System.***out***.println();

}

}

}

**Expected I/P:**

Enter the number of rows : 5

**Expected O/P:**

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

\*\*\*\*

\*\*\*

\*\*

\*

1. **Write a program for Half Diamond pattern using stars and numbers in a palindromic pattern?**

**Algorithm:**

step 1: START

step 2: Initialize variables

step 3: Check for loop condition.If true, goto step 4 else goto step 11

step 4: Check for loop condition.If true,goto step 5 else goto step 8

step 5: Check if statement condition.If true,goto step 6 else goto step 7

step 6: Print "\*" and goto step 7

step 7: Print j value and repeat step 4

step 8: Check for loop condition.If true,goto step 9 else goto step 10

step 9: Print j value and repeat step 8

step 10: Print "\*" and print on next line.Repeat step 3

step 11: Check for loop condition.If true,goto step 12 else goto step 19

step 12: Check for loop condition.If true,goto step 13 else goto step 16

step 13: Check if statement condition.If true,goto step 14 else goto step 15

step 14: Print "\*" and goto step 15

step 15: Print j value and repeat step 12

step 16: Check for loop condition.If true,goto step 17 else goto step 18

step 17: Print j value and repeat step 16

step 18: Print "\*" and print on next line. Repeat step 11

step 19: Print on next line and goto step 20

step 20: STOP

**Program:**

**package** \_6.Patterns\_Programs;

**import** java.util.Scanner;

**public** **class** HalfDiamondstarsandnumbers {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

**int** N, i, j;

Scanner sc = **new** Scanner(System.***in***);

System.***out***.print("\nEnter the number of rows : ");

System.***out***.println();

N = sc.nextInt();

System.***out***.print("\*\n");

**for**(i=1; i<=N; i++)

{

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

{

**if**(j == 1)

System.***out***.print("\*");

System.***out***.print(j);

}

**for**(j=i-1; j>=1; j--)

{

System.***out***.print(j);

}

System.***out***.print("\*");

System.***out***.print("\n");

}

**for**(i=N-1; i>=1; i--)

{

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

{

**if**(j == 1)

System.***out***.print("\*");

System.***out***.print(j);

}

**for**(j=i-1; j>=1; j--)

{

System.***out***.print(j);

}

System.***out***.print("\*");

System.***out***.print("\n");

}

System.***out***.print("\*\n");

}

}

**Expected I/P:**

Enter the number of rows :

4

**Expected O/P:**

\*

\*1\*

\*121\*

\*12321\*

\*1234321\*

\*12321\*

\*121\*

\*1\*

\*

1. **Programs for diamond pattern printing using numbers is discussed here?**

**Algorithm:**

step 1: START

step 2: Initialize variables

step 3: Check for loop condition.If true, goto step 4 else goto step 11

step 4: Check for loop condition.If true,goto step 5 else goto step 6

step 5: Print i value and repeat step 4

step 6: Print in next line and update i value to i+1 and repeat step 3

step 7: Check for loop condition.If true,goto step 8 else goto step 11

step 8: Check for loop condition. If true, goto step 9 else goto step 10

step 9: Print i values and repeat step 8

step 10: Print in next line and Assign i value to i-1.Repeat step 7

step 11: STOP.

**Program:**

**package** \_6.Patterns\_Programs;

**import** java.util.Scanner;

**public** **class** Diamondpatternnumbers {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

Scanner sc = **new** Scanner(System.***in***);

**int** i, j, s, N, count = 0;

s = sc.nextInt();

N = sc.nextInt();

**for** (i = s; count < 4; count++) {

**for** (j = 0; j < count + 1; j++)

System.***out***.print(i);

System.***out***.print("\n");

i = i + 1;

}

**for** (i = s + N - 2; count > 0; count--) {

**for** (j = 0; j < count - 1; j++)

System.***out***.print(i);

System.***out***.print("\n");

i = i - 1;

}

}

}

**Expected I/P:**

4

5

**Expected O/P:**

4

55

666

7777

777

66

5

1. **Write a program for Half diamond using numbers and stars (Printing same numbers)?**

**Algorithm:**

step 1: START

step 2: Initialize variables

step 3: Check for loop condition.If true, goto step 4 else goto step 11

step 4: Assign k value to 1 and goto step 5

step 5: Check for loop condition.If true,goto step 6 else goto step 10

step 6: Print i values and goto step 7

step 7: Check if statement condition.If true,goto step 8 else goto step 10

step 8: Print "\*" and goto step 9

step 9: Update k value to k+1 and repeat step 5

step 10: Print on next line and repeat step 3

step 11: Check for loop condition.If true,goto step 12 else goto step

step 12: Assign k value to 1 and goto step 13

step 13: Check for loop condition.If true,goto step 14 else goto step

step 14: Print i value and goto step 15

step 15: Check if statement condition.If true,goto step 15 else goto step 17

step 16: Print "\*" and update k value to k+1 and repeat step 13

step 17: Print on next line and repeat step 11

step 18: STOP

**Program:**

**package** \_6.Patterns\_Programs;

**import** java.util.Scanner;

**public** **class** Halfdiamondnumbers {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

Scanner sc = **new** Scanner(System.***in***);

**int** i, j, k, N, count = 0;

N = sc.nextInt();

**for** (i = 1; i <= N; i++) {

k = 1;

**for** (j = 0; j < i; j++) {

System.***out***.print(i);

**if** (k < i) {

System.***out***.print("\*");

k = k + 1;

}

}

System.***out***.print("\n");

}

**for** (i = N; i > 0; i--) {

k = 1;

**for** (j = 0; j < i; j++) {

System.***out***.print(i);

**if** (k < i) {

System.***out***.print("\*");

k = k + 1;

}

}

System.***out***.println();

}

}

}

**Expected I/P:**

4

**Expected O/P:**

1

2\*2

3\*3\*3

4\*4\*4\*4

4\*4\*4\*4

3\*3\*3

2\*2

1

1. **Write a program for Half diamond pattern using numbers and stars (different numbers in a line)?**

**Algorithm:**

step 1: START

step 2: Initialize variables

step 3: Check for loop condition.If true, goto step 4 else goto step 9

step 4: Check for loop condition.If true,goto step 5 else goto step 8

step 5: Check if statement condition.If true,goto step 6 else goto step 7

step 6: Print increment of count and "\*".Repeat step 4

step 7: Increment count and print result.Repeat step 4

step 8: Print on next line and repeat step 3

step 9: Update count to count-n and goto step 10

step 10: Check for loop condition.If true,goto step 11 else goto step 16

step 11: Check for loop condition.If true,goto step 12 else goto step 15

step 12: Check if statement condition.If true,goto step 13 else goto step 14

step 13: Increment count and print both count and "\*".Repeat step 11

step 14: Increment count and print the result.Repeat step 11

step 15: Update count value to (count+1)-2\*i and print in next line.Repeat step 10

step 16: STOP

**Program:**

**package** \_6.Patterns\_Programs;

**import** java.util.Scanner;

**public** **class** Halfdiamondpatternnumbers {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

Scanner sc = **new** Scanner(System.***in***);

**int** i, j, count = 1, n;

n = sc.nextInt();

**for** (i = 1; i <= n; i++) {

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

**if** (j < i)

System.***out***.print(count++ + "\*");

**else**

System.***out***.print(count++);

}

System.***out***.print("\n");

}

count = count - n;

**for** (i = n; i >= 1; i--) {

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

**if** (j < i)

System.***out***.print(count++ + "\*");

**else**

System.***out***.print(count++);

}

count = (count + 1) - 2 \* i;

System.***out***.println();

}

}

}

**Input:**

6

**Expected O/P**

1

2\*3

4\*5\*6

7\*8\*9\*10

11\*12\*13\*14\*15

16\*17\*18\*19\*20\*21

16\*17\*18\*19\*20\*21

11\*12\*13\*14\*15

7\*8\*9\*10

4\*5\*6

2\*3

1

1. **Write a program for Half Diamond pattern using stars and numbers in a palindromic pattern?**

**Algorithm:**

step 1: START

step 2: Initialize variables

step 3: Check for loop condition.If true, goto step 4 else goto step 11

step 4: Check for loop condition.If true,goto step 5 else goto step 8

step 5: Check if statement condition.If true,goto step 6 else goto step 7

step 6: Print "\*" and goto step 7

step 7: Print j values and repeat step 4

step 8: Check for loop condition.If true,goto step 9 else goto step 10

step 9: Print j values and repeat step 8

step 10:Print "\*" and print in next line.Repeat step 3

step 11:Check for loop condition.If true,goto step 12 else goto step 19

step 12: Check for loop condition.If true,goto step 13 else goto step 16

step 13: Check if statement condition.If true,goto step 14 else goto step 15

step 14: Print "\*" and goto step 16

step 15: Print j value and repeat step 12

step 16: Check for loop condition.If true,goto step 17 else goto step 18

step 17: Print j value and repeat step 16

step 18: Print "\*" and print in next line.Repeat step 12

step 19: Print on next line and goto step 20

step 20: STOP.

**Program:**

**package** \_6.Patterns\_Programs;

**import** java.util.Scanner;

**public** **class** HalfDiamondnumbers\_1 {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

**int** N, i, j;

Scanner sc = **new** Scanner(System.***in***);

System.***out***.print("\nEnter the number of rows : ");

System.***out***.println();

N = sc.nextInt();

System.***out***.print("\*\n");

**for** (i = 1; i <= N; i++) {

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

**if** (j == 1)

System.***out***.print("\*");

System.***out***.print(j);

}

**for** (j = i - 1; j >= 1; j--) {

System.***out***.print(j);

}

System.***out***.print("\*");

System.***out***.print("\n");

}

**for** (i = N - 1; i >= 1; i--) {

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

**if** (j == 1)

System.***out***.print("\*");

System.***out***.print(j);

}

**for** (j = i - 1; j >= 1; j--) {

System.***out***.print(j);

}

System.***out***.print("\*");

System.***out***.print("\n");

}

System.***out***.print("\*\n");

}

}

**Input:**

Enter the number of rows :

3

**Expected O/P:**

\*

\*1\*

\*121\*

\*12321\*

\*121\*

\*1\*

\*

1. **Program to print Pascal's triangle is discussed here. Pascals triangle is a triangular array of the binomial coefficients?**

The numbers outside Pascal's triangle are all "0". These "0s" are very important for the triangular pattern to work to form a triangular array. The triangle starts with a number "1" above, and any new number added below the upper number "1" is just the sum of the two numbers above, except for the edge, which is all "1".

**pascal's triangle**

row 0 =1

row 1 = (0+1), (1+0) = 1, 1

row 2 = (0+1), (1+1), (1+0) = 1, 2, 1

row 3 = (0+1), (1+2), (2+1), (1+0) = 1, 3, 3, 1

row 4 = (0+1), (1+3), (3+3), (3+1), (1+0) = 1, 4, 6, 4, 1

row 5 = (0+1), (1+4), (4+6), (6+4), (4+1),(1+0) = 1, 5, 10, 10, 5, 1

row 6 = (0+1), (1+5), (5+10), (10+10), (10+5), (5+1), (1+0) = 1, 6, 15, 20, 15, 6, 1

**Algorithm:**

step 1: START

step 2: Initialize variables

step 3: Check for loop condition.If true, goto step 4 else goto step 12

step 4: Check for loop condition.If true,goto step 5 else goto step 6

step 5: Print space and repeat step 4

step 6: Check for loop condition.If true,goto step 7 else goto step 11

step 7: Check if statement condition.If true,goto step 8 else goto step 9

step 8: Assign coef to 1 and goto step 10

step 9: Assign coef to coef\*(i-j+1)/j and goto step 10

step 10: Print coef and repeat step 6

step 11: Print on next line and repeat step 3

step 12: STOP.

**Program:**

**package** \_6.Patterns\_Programs;

**import** java.util.Scanner;

**public** **class** Pascalstriangle {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

**int** i,j, space;

**int** coef, rows;

coef = 1;

Scanner sc = **new** Scanner(System.***in***);

System.***out***.print("Enter the number of rows : ");

rows = sc.nextInt();

**for**(i=0; i<rows; i++)

{

**for**(space=1; space <= rows-i; space++)

System.***out***.print(" ") ;

**for**(j=0; j <= i; j++)

{

**if** (j==0 || i==0)

coef = 1;

**else**

coef = coef\*(i-j+1)/j;

System.***out***.print(coef + " ");

}

System.***out***.println("\n");

}

}

}

**Input:**

Enter the number of rows : 6

**Expected O/P:**

1

1 1

1 2 1

1 3 3 1

1 4 6 4 1

1 5 10 10 5 1

1. **Write a program for Hollow Diamond Inscribed in a Rectangle?**

**Algorithm:**

step 1: START

step 2: Initialize variables

step 3: Check for loop condition.If true, goto step 4 else goto step 12

step 4: Check for loop condition.If true,goto step 5 else goto step

step 5:Check if statement condition.If true,goto step 6 else goto step 7

step 6: Print "\*" and repeat step 4

step 7: Print space and repeat step 4

step 8: Check if statement condition.If true,goto step 9 else goto step 10

step 9: Print "\*" and repeat step 4

step 10: Print space and repeat step 4

step 11: Print in next line and repeat step 3

step 12: Check for loop condition.If true,goto step 13 else goto step 21

step 13: Check for loop condition.If true,goto step 14 else goto step 20

step 14: Check if statement condition.If true,goto step 15 else goto step 16

step 15: Print "\*" and repeat step 13

step 16: Print space and repeat step 13

step 17: Check if statement condition.If true,goto step 18 else goto step 19

step 18: Print "\*" and repeat step 13

step 19: Print space and repeat step 13

step 20: Print in next line and repeat step 12

step 21: STOP.

**Program:**

**package** \_6.Patterns\_Programs;

**import** java.util.Scanner;

**public** **class** HollowDiamondInscribed {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

Scanner sc = **new** Scanner(System.***in***);

**int** i, j;

**int** n = sc.nextInt();

// upper half of the pattern

**for** (i = 0; i < n; i++) {

**for** (j = 0; j < (2 \* n); j++) {

**if** (i + j <= n - 1) // upper left triangle

System.***out***.print("\*");

**else**

System.***out***.print(" ");

**if** ((i + n) <= j) // upper right triangle

System.***out***.print("\*");

**else**

System.***out***.print(" ");

}

System.***out***.println();

}

// bottom half of the pattern

**for** (i = 0; i < n; i++) {

**for** (j = 0; j < (2 \* n); j++) {

**if** (i >= j) // bottom left triangle

System.***out***.print("\*");

**else**

System.***out***.print(" ");

**if** (i >= (2 \* n - 1) - j) // bottom right triangle

System.***out***.print("\*");

**else**

System.***out***.print(" ");

}

System.***out***.println();

}

}

}

**Input:**

4

**Expected O/P:**

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

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

\* \* \* \*

\* \*

\* \*

\* \* \* \*

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

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

1. **Write a program for Butterfly Pattern Printing?**

**Algorithm:**

step 1: START

step 2: Initialize variables

step 3: Check for loop condition.If true, goto step 4 else goto step 12

step 4: Check for loop condition.If true,goto step 5 else goto step 11

step 5:Check if statement condition.If true,goto step 6 else goto step 7

step 6: Print "\*" and repeat step 4

step 7: Print space and repeat step 4

step 8: Check if statement condition.If true,goto step 9 else goto step 10

step 9: Print "\*" and repeat step 4

step 10: Print space repeat step 4

step 11: Print in next line and repeat step 3

step 12: Check for loop condition.If true,goto step 13 else goto step 21

step 13: Check for loop condition.If true,goto step 14 else goto step 20

step 14: Check if statement condition.If true,goto step 15 else goto step 16

step 15: Print "\*" and repeat step 13

step 16: Print space and repeat step 13

step 17: Check if statement condition.If true,goto step 18 else goto step 19

step 18: Print "\*" and repeat step 13

step 19: Print space and repeat step 13

step 20: Print in next line and repeat step 12.

step 21: STOP.

**Program:**

**package** \_6.Patterns\_Programs;

**import** java.util.Scanner;

**public** **class** ButterflyPattern {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

Scanner sc = **new** Scanner(System.***in***);

**int** i, j;

**int** n = sc.nextInt();

// upper half of the pattern

**for**(i = 0; i < n; i++)

{

**for**(j = 0; j < (2 \* n); j++)

{

**if**(i >= j) // upper left triangle

System.***out***.print("\*");

**else**

System.***out***.print(" ");

**if**(i >= (2 \* n - 1) - j) // upper right triangle

System.***out***.print("\*");

**else**

System.***out***.print(" ");

}

System.***out***.println();

}

// bottom half of the pattern

**for**(i = 0; i < n; i++)

{

**for**(j = 0; j < (2 \* n); j++)

{

**if**(i + j <= n - 1) // bottom left triangle

System.***out***.print("\*");

**else**

System.***out***.print(" ");

**if**((i + n) <= j) // bottom right triangle

System.***out***.print("\*");

**else**

System.***out***.print(" ");

}

System.***out***.println();

}

}

}

**Input:**

5

**Expected O/P:**

\* \*

\* \* \* \*

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

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

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

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

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

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

\* \* \* \*

\* \*

1. **Write a program for Diagonal & Sides of a Rectangle?**

**Algorithm:**

step 1: START

step 2: Initialize variables

step 3: Check for loop condition.If true, goto step 4 else goto step 9

step 4: Check for loop condition.If true,goto step 5 else goto step 8

step 5:Check if statement condition.If true,goto step 6 else goto step 7

step 6: Print "\*" and repeat step 4

step 7: Print space and repeat step 4

step 8: Print in next line and repeat step 3

step 9: STOP.

**Program:**

**package** \_6.Patterns\_Programs;

**import** java.util.Scanner;

**public** **class** Diagonal {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

Scanner sc = **new** Scanner(System.***in***);

**int** i, j;

**int** n = sc.nextInt(); // 'n' must be odd

**for** (i = 0; i < n; i++) {

**for** (j = 0; j < n; j++) {

// left diagonal, right diagonal, top horizontal, bottom horizontal, left vertical, right vertical

**if** (i == j || i + j == n - 1 || i == 0 || i == n - 1 || j == 0 || j == n - 1)

System.***out***.print("\*");

**else**

System.***out***.print(" ");

}

System.***out***.println();

}

}

}

**Input:**

7

**Expected O/P:**

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

\*\* \*\*

\* \* \* \*

\* \* \*

\* \* \* \*

\*\* \*\*

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

1. **Write a program for Diagonal & Sides of a Rhombus/Diamond?**

**Algorithm:**

step 1: START

step 2: Initialize variables

step 3: Check for loop condition.If true, goto step 4 else goto step 9

step 4: Check for loop condition.If true,goto step 5 else goto step 8

step 5:Check if statement condition.If true,goto step 6 else goto step 7

step 6: Print "\*" and repeat step 4

step 7: Print space and repeat step 3

step 8: Print in next line and goto step 9

step 9: STOP.

**Program:**

**package** \_6.Patterns\_Programs;

**import** java.util.Scanner;

**public** **class** Rhombus {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

Scanner sc = **new** Scanner(System.***in***);

**int** i, j;

**int** n = sc.nextInt(); // ‘n’ must be odd

**int** num1 = n / 2 \* 3;

**for** (i = 0; i < n; i++) {

**for** (j = 0; j < n; j++) {

// center horizontal, center vertical, upper left diagonal, bottom left diagonal, upper right diagonal, bottom right diagonal

**if** (i == n / 2 || j == n / 2 || i + j == n / 2 || i - j == n / 2 || j - i == n / 2 || i + j == num1)

System.***out***.print("\*");

**else**

System.***out***.print(" ");

}

System.***out***.println();

}

}

}

**Input:**

7

**Expected O/P:**

\*

\*\*\*

\* \* \*

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

\* \* \*

\*\*\*

\*

1. **Write a program for Left and Right Arrows?**

**Algorithm:**

step 1: START

step 2: Initialize variables

step 3: Check for loop condition.If true, goto step 4 else goto step 9

step 4: Check for loop condition.If true,goto step 5 else goto step 8

step 5:Check if statement condition.If true,goto step 6 else goto step 7

step 6: Print "\*" and repeat step 4

step 7: Print space and repeat step 4

step 8: Print in next line and repeat step 3

step 9: Check for loop condition.If true,goto step 10 else goto step 15

step 10: Check for loop condition.If true,goto step 11 else goto step 13

step 11: Check if statement condition.If true,goto step 12 else goto step 13

step 12: Print "\*" and repeat step 10

step 13: Print space and repeat step 10

step 14: Print in next line and repeat step 9

step 15: STOP.

**Program:**

**package** \_6.Patterns\_Programs;

**import** java.util.Scanner;

**public** **class** LeftandRightArrows {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

Scanner sc = **new** Scanner(System.***in***);

**int** i, j;

**int** n = sc.nextInt(); // ‘n’ must be odd

**int** num1 = n / 2 \* 3;

// right arrow

System.***out***.println("Right Arrow");

**for** (i = 0; i < n; i++) {

**for** (j = 0; j < n; j++) {

// center horizontal, upper right diagonal, bottom right diagonal

**if** (i == n / 2 || j - i == n / 2 || i + j == num1)

System.***out***.print("\*");

**else**

System.***out***.print(" ");

}

System.***out***.println();

}

// left arrow

System.***out***.println("Left Arrow");

**for** (i = 0; i < n; i++) {

**for** (j = 0; j < n; j++) {

// center horizontal, bottom left diagonal, upper left diagonal

**if** (i == n / 2 || i - j == n / 2 || i + j == n / 2)

System.***out***.print("\*");

**else**

System.***out***.print(" ");

}

System.***out***.println();

}

}

}

**Input:**

5

**Expected O/P:**

Right Arrow

\*

\*

\*\*\*\*\*

\*

\*

Left Arrow

\*

\*

\*\*\*\*\*

\*

\*

1. **Write a program for Rhombus Pattern Program?**

**Algorithm:**

step 1: START

step 2: Initialize variables

step 3: Check for loop condition.If true,goto step 4 else goto step 9

step 4: Check for loop condition.If true,goto step 5 else goto step 6

step 5: Print space and repeat step 4.Goto step 6

step 6: Check for loop condition.If true,goto step 7 else goto step 8

step 7: Print "\*" and repeat step 6.Goto step 8

step 8: Print in next line and repeat step 3.Goto step 9

step 9: Check for loop condition.If true,goto step 10 else goto step

step 10: Check for loop condition.If true,goto step 11 else goto step 12

step 11: Print space and goto step 10.

step 12: Check for loop condition.If true,goto step 13 else goto step 17

step 13: Check if statement condition.If true,goto step 14 else goto step 15

step 14: Print "\*" and goto step 16

step 15: Print space and goto step 16

step 16: Print in next line and repeat step 12

step 17: STOP.

**Program:**

**package** \_6.Patterns\_Programs;

**import** java.util.Scanner;

**public** **class** Rhombus\_1 {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

Scanner sc = **new** Scanner(System.in);

**int** i, j;

**int** n = sc.nextInt();

// solid rhombus

System.out.println("Solid Rhombus");

**for** (i = 0; i < n; i++) {

**for** (j = 0; j < n - i; j++) {

System.out.print(" "); // leading spaces

}

**for** (j = 0; j < n; j++) {

System.out.print("\*");

}

System.out.println();

}

// hollow rhombus

System.out.println("Hollow Rhombus");

**for** (i = 0; i < n; i++) {

**for** (j = 0; j < n - i; j++) {

System.out.print(" "); // leading spaces

**for** (j = 0; j < n; j++) {

// upper horizontal, bottom horizontal, left diagonal, right diagonal

**if** (i == 0 || i == n - 1 || j == 0 || j == n - 1)

System.out.print("\*");

**else**

System.out.print(" ");

}

System.out.println();

}

}

}

**Input:**

4

**Expected O/P:**

Solid Rhombus

\*\*\*\*

\*\*\*\*

\*\*\*\*

\*\*\*\*

Hollow Rhombus

\*\*\*\*

\* \*

\* \*

\*\*\*\*