VELAMMAL ENGINEERING COLLEGE

DEPARTMENT OF ELECTRONICS AND COMMUNICATION

CODING TEST II

Editorial

1. C:

```
/* C program to print solid rectangle star pattern */

#include <stdio.h>

/* Function to print solid rectangle*/

void solid_rectangle(int n, int m)

{

    int i, j;

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

    {

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

        {

            printf(“*”);

        }

        printf(“n”);

    }

}

int main()
```

```
{
    int rows, columns;
    printf(“nEnter the number of rows : “);
    scanf(“%d”, &rows);
    printf(“nEnter the number of columns : “);
    scanf(“%d”, &columns);
    printf(“n”);
    solid_rectangle(rows, columns);
    return 0;
}
```

Python:

```
def solid_rectangle(n, m):
    for i in range(1, n+1):
        for j in range(1, m+1):
            print(“*”, end = “”)
            print()
            rows = int(input(“Enter the number of rows : “))
            columns = int(input(“Enter the number of columns : “))
solid_rectangle(rows, columns)
```

C:

```
#include <stdio.h>
/* Function to print hollow rectangle*/
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)
                printf("*");
            else
                printf(" ");
        }
        printf("n");
    }
}
int main()
{
    int rows, columns;
```

```
    printf(“nEnter the number of rows : “);
    scanf(“%d”, &rows);
    printf(“nEnter the number of columns : “);
    scanf(“%d”, &columns);
    printf(“n”);
    hollow_rectangle(rows, columns);
    return 0;
}
```

Python:

```
def hollow_rectangle(n, m):
    for i in range(1, n+1):
        for j in range(1, m+1):
            if(i == 1 or i == n or j == 1 or j == m):
                print(“*”, end = “”)
                else:
                    print(” “, end = “”)
                    print()

                    rows = int(input(“Enter the number of rows : “))
                    columns = int(input(“Enter the number of columns : “))
hollow_rectangle(rows, columns)
```

2. C:

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

Python:

```
n = int(input())
for i in range(1, n+1) :
    for j in range(1, i+1) :
        print(“*”, end=””)
print()
```

C:

```
#include <stdio.h>
int main()
{
    int i, j, n, k = 0;
    scanf("%d",&n);

    for(i=n; i>=1; --i)
    {
        for(j=0; j < n-i; ++j)
            printf(" ");

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

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

        printf("\n");
    }

    return 0;
}
```

Python:

```
n = int(input())
for i in range(n,1,-1):
    for j in range(0, n-i+1):
        print(" ", end = "")

        for j in range(i, 2*i-1,1):
            print("* ", end = "")

            for j in range(0, i – 1, 1):
                print("* ", end ="")

print("\n")
```

C:

```
#include <stdio.h>
int main()
{
    int n;
    scanf(“%d”,&n);
    int i, j, k = 0;
    for (i = 1; i <= n; i++)
    {
        for (j = i; j < n; j++) {
            printf(” “);
        }
        while (k != (2 * i – 1)) {
            if (k == 0 || k == 2 * i – 2)
                printf(“*”);
            else
                printf(” “);
            k++;
            ;
        }
        k = 0;
        printf(“\n”); // print next row
    }
    for (i = 0; i < 2 * n – 1; i++) {
        printf(“*”);
    }
}
```

3. C:

```
#include<stdio.h>
int main()
{
    int n, i, j, space, count = 1, num = 0, star = 8;
    scanf("%d", &n);
    space = n;
    for (i = 1; i <= n; i++)
    {
        for (j = 1; j <= star; j++)
            if(i + j <= star + 1)
                printf("*");
        num++;
        for (j = 1; j <= i; j++)
        {
            printf("%d", num);
            if (i > 1 && count < i)
            {
                printf("*");
                count++;
            }
        }
        for (j = 1; j <= star; j++)
            if(i + n <= j + n)
                printf("*");
        printf("\n");
        space–;
        count = 1;
    }
    return 0;
}
```

4. C:

```
#include <stdio.h>

int main()
{
    int n, c, k, space = 1;

    printf("Enter the number of rows\n");
    scanf("%d", &n);

    space = n – 1;

    for (k = 1; k <= n; k++)
    {
        for (c = 1; c <= space; c++)
            printf(" ");
        space–;
        for (c = 1; c <= 2*k-1; c++)
            printf("*");
        printf("\n");
    }
    space = 1;

    for (k = 1; k <= n – 1; k++)
    {
        for (c = 1; c <= space; c++)
            printf(" ");
        space++;
        for (c = 1 ; c <= 2*(n-k)-1; c++)
            printf("*");
        printf("\n");
    }
    return 0;
}
```

Python:

```
def Diamond_pattern(rows):
    n = 0
    for i in range(1, rows + 1):
        for j in range (1, (rows – i) + 1):
            print(end = " ")
```

```
            while n != (2 * i – 1):
                print(“*”, end = “”)
                n = n + 1
                n = 0

                print()

                k = 1
                n = 1
                for i in range(1, rows):
                    for j in range (1, k + 1):
                        print(end = ” “)
                        k = k + 1

                        while n <= (2 * (rows – i) – 1):
                            print(“*”, end = “”)
                            n = n + 1
                            n = 1
                            print()

                            rows = int(input(“Enter the number of rows : “))
Diamond_pattern(rows)
```

C:

```
#include <stdio.h>

int main()
{
    int i, j, space, k = 0, n;
    printf(“\nEnter the number of rows : “);
    scanf(“%d”,&n);
    for (i = 1; i <= n; i++)
    {
        for (j = 1; j <= n – i; j++)
        {
            printf(” “);
        }
        while (k != (2 * i – 1))
        {
            if (k == 0 or k == 2 * i – 2)
                printf(“*”);
```

```
            else
                printf(" ");
            k++;
        }
        k = 0;
        printf("\n");
    }
    n–;
    for (i = n; i >= 1; i–)
    {
        for (j = 0; j <= n – i; j++)
        {
            printf(" ");
        }
        k = 0;
        while (k != (2 * i – 1))
        {
            if (k == 0 or k == 2 * i – 2)
                printf("*");
            else
                printf(" ");
            k++;
        }
        printf("\n");
    }
}
```

Python:

```
def Diamond_pattern(n) :
    k = 0;
    for i in range(1,n+1) :
        for j in range(1,n-i+1) :
            print(" ",end="")
            while (k != (2 * i – 1)) :
                if (k == 0 or k == 2 * i – 2) :
                    print("*",end="")
                    else :
                        print(" ",end="")
                        k = k + 1
                        k = 0
                        print("")
```

```
                    n = n – 1

                    for i in range (n,0,-1) :
                        for j in range(0,n-i+1) :
                            print(” “,end=””)
                            k = 0
                            while (k != (2 * i – 1)) :
                                if (k == 0 or k == 2 * i – 2) :
                                    print(“*”,end=””)
                                    else :
                                        print(” “,end=””)
                                        k = k + 1
                                        print(“”)

                                        n = int(input(“Enter the number of rows : “))
Diamond_pattern(n)
```

C:

```
#include <stdio.h>

int main()
{
    int i, j, space, k = 0, n;
    printf(“\nEnter the number of rows : “);
    scanf(“%d”,&n);
    for(int i=1;i<=n;i++)
    {
        for(int j=1;j<=n-i;j++)
        {
            printf(” “);
        }
        for(int j=1;j<=i;j++)
        {
            printf(“*”);
        }
        printf(“\n”);
    }
    for(int i=n-1;i>0;i–)
    {
        for(int j=1;j<=n-i;j++)
```

```
      {
         printf(" ");
      }
      for(int j=1;j<=i;j++)
      {
         printf("*");
      }
      printf("\n");
   }
}
```

5. C:

```
#include <stdio.h>
int main()
{
    int i, j, n;
    scanf("%d", &n);
    // 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
                printf("*");
            else
                printf(" ");
            if((i + n) <= j)  // upper right triangle
                printf("*");
            else
                printf(" ");
        }
        printf("\n");
    }
    // bottom half of the pattern
    for(i = 0; i < n; i++)
    {
        for(j = 0; j < (2 * n); j++)
        {
            if(i >= j)  //bottom left triangle
                printf("*");
            else
                printf(" ");
            if(i >= (2 * n - 1) - j)  // bottom right
triangle
                printf("*");
            else
                printf(" ");
```

```
        }
        printf("\n");
    }
    return 0;
}
```

6. C:

```
#include <stdio.h>
int main()
{
    int i, j, n;
    scanf("%d", &n);
    // upper half of the pattern
    for(i = 0; i < n; i++)
    {
        for(j = 0; j < (2 * n); j++)
        {
            if(i >= j)  // upper left triangle
                printf("*");
            else
                printf(" ");
            if(i >= (2 * n - 1) - j) // upper right triangle
                printf("*");
            else
                printf(" ");
        }
        printf("\n");
    }
    // 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
                printf("*");
            else
                printf(" ");
            if((i + n) <= j) // bottom right triangle
                printf("*");
            else
                printf(" ");
        }
        printf("\n");
    }
    return 0;
}
```

7. C:

```
#include <stdio.h>
int main()
{
    int i, j, n;
    scanf("%d", &n);  // '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)
                printf("*");
            else
                printf(" ");
        }
        printf("\n");
    }
    return 0;
}
```

8. C:

```
#include <stdio.h>
int main()
{
int i, j, n;
scanf("%d", &n);  // ‘n’ must be odd
int num1 = n / 2 * 3;
// right arrow
printf("Right Arrow\n");
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)
printf("*");
else
printf(" ");
}
printf("\n");
}
// left arrow
printf("Left Arrow\n");
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)
printf("*");
else
printf(" ");
}
printf("\n");
}
return 0;
}
```

9. C:

```
#include<stdio.h>
#include<string.h>
int min(int num1,int num2)
{
	return (num1>num2)?num2:num1;
}
int main()
{
	char a[50];
	int r,c,i,j;
	scanf("%s",a);
	r=(strlen(a))*2-1;
	c=r;
	for (i=0;i<r;i++){
	    for (j=0;j<c-1;j++){

	        if ((i+j)>=(r-1))
	            printf("%c",a[min(((r-1)-i),((i+j)-(r-1)))]);
	        else
	            printf(" ");}
	    for (j=0;j<c;j++){

	        if (i>=j)
	            printf("%c",a[min((r-1)-i,(i*2)-(i+j))]);
	        else
	            printf(" ");}
	    printf("\n");}
	    return 0;
```

```
}
```

Python:

```
a=input()
r=len(a)*2-1
c=r
for i in range(r):
    for j in range(c-1):
        if (i+j)>=(r-1):
            x=min(((r-1)-i),((i+j)-(r-1)))
            print(a[x],end="")
        else:
            print(end=" ")
    for j in range(c):
        if i>=j:
            print(a[min((r-1)-i,(i*2)-(i+j))],end="")
        else:
            print(end=" ")
    print()
```

10. Solution:

```
for i in range(1,int(input())):
        print(((10**i-1)//9)*i)
```