

NAME: Amrit Tamang

ROLL NO: THA080BEI006

Level 4: Mastermind

1. Write a C program to make such a pattern like a pyramid with a number which will repeat the number in the same row.

```
1
2 2
3 3 3
4 4 4 4
```

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    int n;
```

```
    printf("Enter the number of rows: ");
```

```
    scanf("%d", &n);
```

```
    for (int i = 0; i < n; i++)
```

```
    {
```

```
        for (int j = 0; j < n - i; j++) // for space
```

```
        {
```

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

```
        }
```

```
        for (int k = 0; k < i + 1; k++) // for number pattern
```

```
        {
```

```
            printf("%d ", i + 1);
```

```
        }
```

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

```

    }
    return 0;
}

```

OUTPUT:

```

{ .\01_assignment_4 }
Enter the number of rows: 4
  1
 2 2
3 3 3
4 4 4 4

```

2. Write a program in C to display a pattern like a diamond.

```

    *
   ***
  *****
 *****
*****
 *****
  *****
   ***
    *

```

```

#include <stdio.h>

int main()
{
    int noOfSpace = 8 / 2; // initial number of space
    int noOfStars = 1;    // initial number of stars
    int middleLine = 8 / 2; // middle line so we could reverse the condition of upper part
    in lower part
    for (int i = 0; i <= 8; i++)
    {
        for (int j = 0; j < noOfSpace; j++)

```

```

    {
        printf(" ");
    }
    for (int k = 0; k < noOfStars; k++)
    {
        printf("*");
    }
    printf("\n");
    if (i < middleLine)
    {
        noOfSpace--;
        noOfStars += 2;
    }
    else
    {
        noOfSpace++;
        noOfStars -= 2;
    }
}
return 0;}

```

OUTPUT:

```

nment_4.c -o 02_assignment_4 } ; if ($?) { .\02_assignment_4 }
    *
    ***
    *****
    *********
    *********
    *****
    *****
    ***
    *

```

3. Write a C program to display a such a pattern for n rows using a number that starts with 1 and each row will have a 1 as the first and last number.

```
1
121
12321
```

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    int n;
```

```
    printf("Enter the number of rows: ");
```

```
    scanf("%d", &n);
```

```
    for (int i = 0; i < n; i++)
```

```
    {
```

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

```
        {
```

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

```
        }
```

```
        for (int k = 0; k <= i; k++)
```

```
        {
```

```
            printf("%d", k + 1);
```

```
        }
```

```
        for (int l = i; l > 0; l--)
```

```
        {
```

```
            printf("%d", l);
```

```
        }
```

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

```
    }
```

```
    return 0;
```

```
}
```

OUTPUT:

```
nment_4.c -o 03_assignment_4 } ; if ($?) { .\03_assignment_4 }  
Enter the number of rows: 5  
    1  
   121  
  12321  
 1234321  
123454321
```

4. Write a program in C to convert an octal number into binary.

Test Data :

Input an octal number (using digit 0 - 7) :57

Expected Output :

The Octal Number : 57

The equivalent Binary Number : 101111

```
#include <stdio.h>
```

```
#include <math.h>
```

```
int main()
```

```
{
```

```
    // to convert octal to binary first convert the octal number to decimal and the decimal  
    number into binary
```

```
    int rem;
```

```
    int octalNumber, decimalNumber = 0;
```

```
    int binaryNumber = 0, place = 0;
```

```
    printf("Input an octal number (using digit 0 - 7): ");
```

```
    scanf("%d", &octalNumber);
```

```
    // while loop will convert the octal number into decimal
```

```
    while (octalNumber != 0)
```

```
    {
```

```
        rem = octalNumber % 10;
```

```

        decimalNumber = decimalNumber + rem * pow(8, place);
        octalNumber = octalNumber / 10;
        place++;
    }
    rem = 0;
    place = 1;
    while (decimalNumber != 0)
    {
        rem = decimalNumber % 2;
        binaryNumber = binaryNumber + rem * place;
        decimalNumber = decimalNumber / 2;
        place = place * 10;
    }
    printf("the equivalent Binary Number: %d", binaryNumber);
    return 0;
}

```

OUTPUT:

```

nment_4.c -o 04_assignment_4 } ; if ($?) { .\04_assignment_4 }
Input an octal number (using digit 0 - 7): 57
the equivalent Binary Number: 101111

```

5. Write a C program to display Pascal's triangle.

Test Data :

Input number of rows: 4

Expected Output :

```

    1
   1 1
  1 2 1
 1 3 3 1

```

```
#include <stdio.h>
```

```
int main()
```

```
{
    int n;
    printf("Enter the no of rows: ");
    scanf("%d", &n);
    int value = 0, j, i, k;
    for (i = 1; i <= n; i++)
    {
        for (j = 1; j <= n - i; j++)
        {
            printf(" ");
        }
        for (k = 1; k <= i; k++)
        {
            if (i == 1 || k == 1)
            {
                value = 1;
            }
            else
            {
                value = value * (i - k + 1) / (k - 1);
            }
            printf("%2d", value);
        }
        printf("\n");
    }
    return 0;
}
```

OUTPUT:

```
nment_4.c -o 05_assignment_4 } ; if ($?) { .\05_assignment_4 }  
Enter the no of rows: 5  
    1  
  1 1  
1 2 1  
1 3 3 1  
1 4 6 4 1
```

6. Write a program in C to display a pattern like

a. *

```
  **  
 ***  
  *  
 **  
 ***
```

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    for (int i = 0; i < 3; i++)
```

```
    {
```

```
        for (int j = 0; j <= i; j++)
```

```
        {
```

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

```
        }
```

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

```
    }
```

```
    for (int i = 0; i < 3; i++)
```

```
    {
```

```
        for (int j = 0; j <= i; j++)
```



```

    {
        printf("*");
    }
    printf("\n");
}
return 0;
}

```

OUTPUT:

```

assignment_4.c -o 06_a_assignment_4 } ; if ($?) { .\06_a_assignment_4 }
*
**
***
*
**
***

```

b. @ @
 @ @
 @ @
 @

```

#include <stdio.h>

int main()
{
    int nsp = 5;
    for (int i = 0; i < 4; i++)
    {
        for (int j = 0; j < i; j++)
        {

```

```
        printf(" ");
    }
    for (int k = 0; k < 7; k++)
    {
        if ((i + k) == 2 * i)
        {
            printf("@");
        }
    }
    for (int l = 0; l < nsp; l++)
    {
        printf(" ");
    }
    nsp -= 2;

    for (int k = 0; k < 3; k++)
    {
        if ((i + k) == 2 * i)
        {
            printf("@");
        }
    }
    printf("\n");
}
return 0;
}
```

OUTPUT:

```
ignment_4.c -o 06_b_assignment_4 } ; if ($?) { .\06_b_assignment_4 }
```

```
@      @  
@      @  
@  @  
@
```