Q1.Write a recursive function to print first N natural numbers

```c
#include <stdio.h>

int rec(int);

int main(int argc, char *argv[])
{
    printf("%d ",rec(10));
    return 0;
}

int rec(int num)
{
    if (num == 1)
        return num;
    else
    {
        printf("%d ",rec(num-1));
        return num;
    }
}
```

Q2.Write a recursive function to print first N natural numbers in reverse order

```c
#include <stdio.h>

void rec(int);

int main(int argc, char *argv[])
{
    rec(10);
    return 0;
}

void rec(int num)
{
    if (num == 1)
        printf("%d ", num);
    else
    {
        printf("%d ", num);
        rec(num - 1);
    }
}
```

Q3.Write a recursive function to print first N odd natural numbers

```c
#include <stdio.h>

void rec(int, int);

int main(int argc, char *argv[])
{
    int num;

    printf("Enter number = ");
    scanf("%d", &num);

    rec(num, 1);

    return 0;
}

void rec(int num, int odd)
{
    if (num == 0)
        return;

    printf("%d ", odd);
    rec(num - 1, odd + 2);
}
```

Q4.Write a recursive function to print first N odd natural numbers in reverse order

```c
#include <stdio.h>

void rec(int, int);

int main(int argc, char *argv[])
{
    int num;

    printf("Enter number = ");
    scanf("%d", &num);

    rec(num, num * 2);

    return 0;
}

void rec(int num, int odd)
{
    if (num == 0)
        return;

    else if (odd % 2 != 0)
    {
        printf("%d ", odd);
        num = num - 1;
    }
    rec(num, odd - 1);
    return;
}
```

Q5.Write a recursive function to print first N even natural numbers

```c
#include <stdio.h>

void rec(int, int);

int main(int argc, char *argv[])
{
    rec(15, 2);

    return 0;
}

void rec(int num, int even)
{
    if (num == 0)
        return;

    printf("%d ", even);
    rec(num - 1, even + 2);
}
```

Q6.Write a recursive function to print first N even natural numbers in reverse order

```c
#include <stdio.h>

void rec(int, int);

int main(int argc, char *argv[])
{
    int num;

    printf("Enter number = ");
    scanf("%d", &num);

    rec(num, num * 2);

    return 0;
}

void rec(int num, int even)
{
    if (num == 0)
        return;

    if (even % 2 == 0)
    {
        printf("%d ", even);
        num = num - 1;
    }

    rec(num, even - 1);
}
```

Q7.Write a recursive function to print squares of first N natural numbers

```c
#include <stdio.h>

int rec(int);

int main(int argc, char *argv[])
{
    int num;

    printf("Enter number = ");
    scanf("%d",&num);

    printf("%d = %d",rec(num), (num*num));

    return 0;
}

int rec(int num)
{
    if (num == 1)
        return num;

    int ans = rec(num - 1);
    printf("%d = %d \n", ans, (ans * ans));

    return num;
}
```

Q8.Write a recursive function to print binary of a given decimal number

```c
#include <stdio.h>

void rec(int);

int main(int argc, char *argv[])
{
    int num;

    printf("Enter number = ");
    scanf("%d",&num);

    rec(num);

    return 0;
}

void rec(int num)
{
    if (num == 0)
        return;

    int bin = num % 2;
    rec(num / 2);

    printf("%d", bin);
}
```

Q9.Write a recursive function to print octal of a given decimal number

```c
#include <stdio.h>

void rec(int);

int main(int argc, char *argv[])
{
    int num;

    printf("Enter number = ");
    scanf("%d",&num);

    rec(num);

    return 0;
}

void rec(int num)
{
    if (num == 0)
        return;

    int oct = num % 8;
    rec(num / 8);

    printf("%d", oct);
}
```

Q10.Write a recursive function to print reverse of a given number

```c
#include <stdio.h>

void rec(int num);

int main(int argc, char *argv[])
{
    rec(12045);

    return 0;
}

void rec(int num)
{
    if (num==0)
    {
        return;
    }
    printf("%d",num%10);
    rec(num/10);
    return;
}
```