**Longest substring:**

int lengthOfLongestSubstring(char \* s){

int arr[128] = {0}, left = 0, max = 0;

for(int right = 0; s[right] != '\0'; right++){

if( arr[s[right]] != 0 && arr[s[right]] > left ){

left = arr[s[right]];

}

arr[s[right]] = right + 1;

if ( right - left + 1 > max ){

max = right - left + 1;

}

}

return max;

}

**Happy Number:**

int getNext(int n) {

int res = 0;

while (n) {

res += (n % 10) \* (n % 10);

n /= 10;

}

return res;

}

int isHappy(int n) {

int a = n;

int b = getNext(n);

while (a != b) {

a = getNext(a);

b = getNext(getNext(b));

}

return b == 1;

}

**Printing Patterns Using Loops**:

int n;

scanf("%d", &n);

int len = 2\*n - 1;

for (int i = 0; i < len; i++) {

for (int j = 0; j < len; j++) {

int min = i < j ? i : j;

min = min < len-i ? min : len-i-1;

min = min < len-j-1 ? min : len-j-1;

printf("%d ", n-min);

}

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

**Correctness and Loop variant:**

**void** insertionSort(**int** N, **int** arr[]) {

**int** i,j;

**int** value;

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

    {

        value=arr[i];

        j=i-1;

**while**(j>=0 && value<arr[j])

        {

            arr[j+1]=arr[j];

            j=j-1;

        }

        arr[j+1]=value;

    }

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

    {

        printf("%d",arr[j]);

        printf(" ");

    }

}

**Small triangle and large triangle:**

**float** calcArea (triangle tr) *//func for cleaner code to calculate the area*

{ **float** p = (tr.a + tr.b + tr.c) / 2.0;

**return** sqrt (p \* (p - tr.a) \* (p - tr.b) \* (p - tr.c));

}

**void** sort\_by\_area(triangle\* tr, **int** n)

{

**for** (**int** i = 0; i < n - 1; i++)

{

**for** (**int** j = 0; j - n - 1; j++)

    {

**if**(calcArea(tr[j]) > calcArea(tr[j+1]))

        {

            triangle temp = tr[j];

            tr[j] = tr[j+1];

            tr[j+1] = temp;

        }

    }

}

}

**Triangle numbers:**

int solve(int n) {

if (n < 3)

return -1;

else if (n % 4 == 1 || n % 4 == 3)

return 2;

else if (n % 4 == 0)

return 3;

else

return 4;

}

**Mod 3 For loop in c:**

for(i=a;i<=b;i++)

{

if(i<10)

{

if(i==1)

printf("one\n");

else if(i==2)

printf("two\n");

else if(i==3)

printf("three\n");

else if(i==4)

printf("four\n");

else if(i==5)

printf("five\n");

else if(i==6)

printf("six\n");

else if(i==7)

printf("seven\n");

else if(i==8)

printf("eight\n");

else if(i==9)

printf("nine\n");

}

else {

if(i%2==1)

printf("odd\n");

else {

printf("even\n");

}

}

}

**Nth Tribonacci number:**

**int tribonacci(int n) {**

**int i;**

**// Base cases**

**if (n == 0) return 0;**

**if (n == 1 || n == 2) return 1;**

**// Variables to store Tn-3, Tn-2, and Tn-1**

**int Tn\_3 = 0, Tn\_2 = 1, Tn\_1 = 1;**

**int Tn = 0;**

**// Compute Tn for n >= 3**

**for (i = 3; i <= n; i++) {**

**Tn = Tn\_3 + Tn\_2 + Tn\_1; // Compute Tn**

**// Update Tn-3, Tn-2, Tn-1 for next iteration**

**Tn\_3 = Tn\_2;**

**Tn\_2 = Tn\_1;**

**Tn\_1 = Tn;**

**}**

**return Tn;**

**}**