1.There's a staircase with N steps, and you can climb 1 or 2 steps at a time. Given N, write a function that returns the number of unique ways you can climb the staircase. The order of the steps matters.

For example, if N is 4, then there are 5 unique ways:

1, 1, 1, 1

2, 1, 1

1, 2, 1

1, 1, 2

2, 2

What if, instead of being able to climb 1 or 2 steps at a time, you could climb any number from a set of positive integers X? For example, if X = {1, 3, 5}, you could climb 1, 3, or 5 steps at a time. Generalize your function to take in X.

#include <stdio.h>

  int findStep(int n)

{

    if (n == 1 || n == 0)

        return 1;

    else if (n == 2)

        return 2;

    else

        return findStep(n - 3) + findStep(n - 2) + findStep(n - 1);

}

int main()

{

    int n = 4;

    printf("%d\n", findStep(n));

    return 0;

}

2. How do you check if two rectangles overlap with each other

class Point:

    def \_\_init\_\_(self, x, y):

        self.x = x

        self.y = y

def doOverlap(l1, r1, l2, r2):

    if(l1.x >= r2.x or l2.x >= r1.x):

        return False

   if(l1.y <= r2.y or l2.y <= r1.y):

        return False

   return True

  if \_\_name\_\_ == "\_\_main\_\_":

    l1 = Point(0, 10)

    r1 = Point(10, 0)

    l2 = Point(5, 5)

    r2 = Point(15, 0)

    if(doOverlap(l1, r1, l2, r2)):

        print("Rectangles Overlap")

    else:

        print("Rectangles Don't Overlap")

3. How to Find Missing Number on Integer Array of 1 to 100

#include <stdio.h>

int getMissingNo(int a[], int n)

{

    int i, total;

    total = (n + 1) \* (n + 2) / 2;

    for (i = 0; i < n; i++)

        total -= a[i];

    return total;

}

int main()

{

    int a[] = { 1, 2, 4, 5, 6 };

    int miss = getMissingNo(a, 5);

    printf("%d", miss);

    getchar();

}