TRIANGLE TRIPLETS EDITORIAL

Since N <= 100, it is possible to try out all the tuples from the given sticks and check if it satisfy the condition of being a scalene triangle. For this, we can use three for loops.

For a, b and c to be the sides of a triangle it must satisfy the following properties :

1. The sum of any two sides must be greater than third side.
2. The absolute difference of any two sides must be less than the third side.

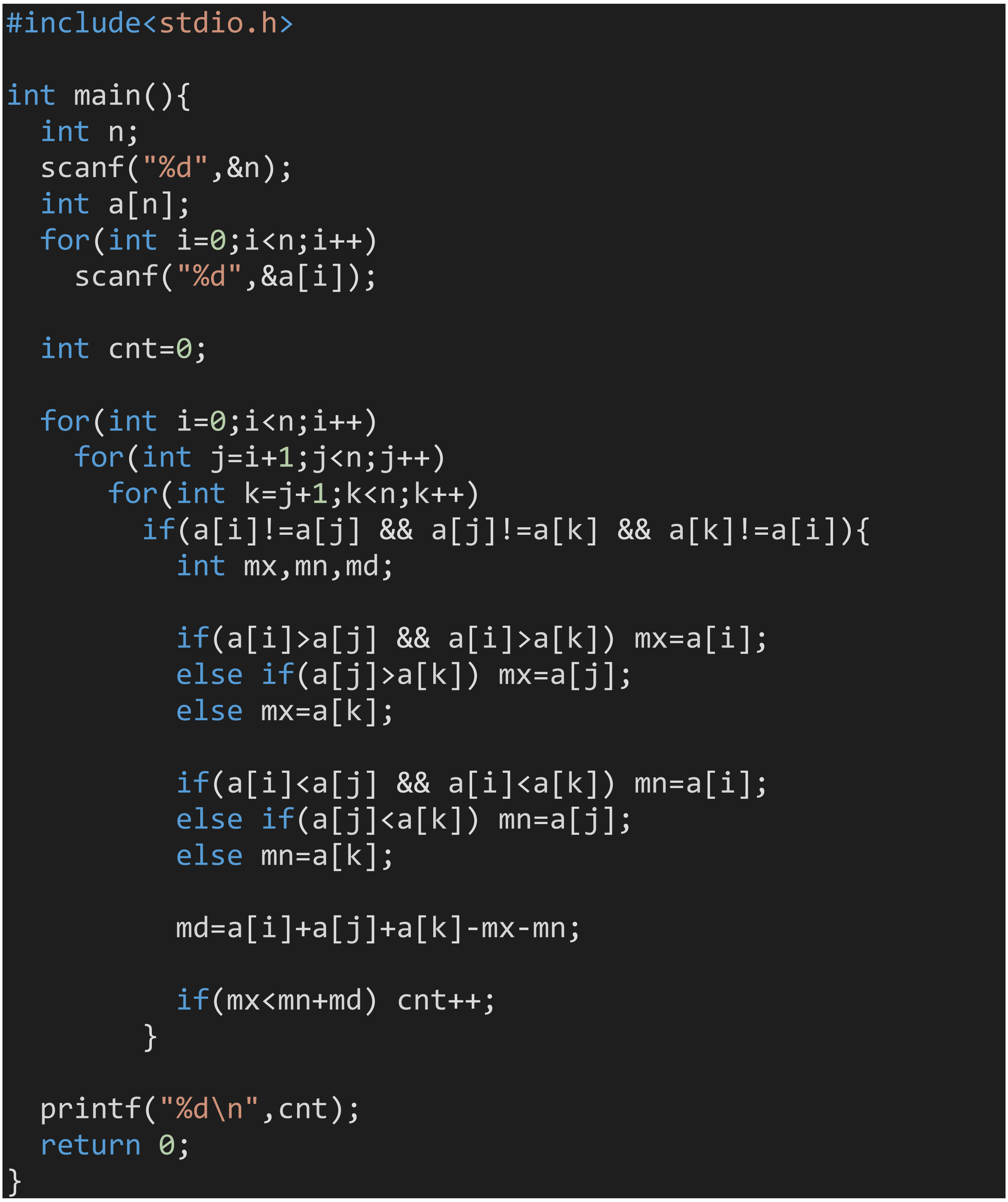
Both these rules can be checked using only one condition that is if x=max(a,b,c) and z=min(a,b,c) and y is the middle element, then x < (y+z).

Also, for a triangle to be scalene all its sides must be having distinct lengths. That is if a, b and c are the sides of a triangle, the a!=b and b!=c and c!=a.

Time Complexity = $O(N^3)$

Space Complexity = O(1)

CODE:



```c
#include<stdio.h>

int main(){
  int n;
  scanf("%d",&n);
  int a[n];
  for(int i=0;i<n;i++)
    scanf("%d",&a[i]);

  int cnt=0;

  for(int i=0;i<n;i++)
    for(int j=i+1;j<n;j++)
      for(int k=j+1;k<n;k++)
        if(a[i]!=a[j] && a[j]!=a[k] && a[k]!=a[i]){
          int mx,mn,md;

          if(a[i]>a[j] && a[i]>a[k]) mx=a[i];
          else if(a[j]>a[k]) mx=a[j];
          else mx=a[k];

          if(a[i]<a[j] && a[i]<a[k]) mn=a[i];
          else if(a[j]<a[k]) mn=a[j];
          else mn=a[k];

          md=a[i]+a[j]+a[k]-mx-mn;

          if(mx<mn+md) cnt++;
        }

  printf("%d\n",cnt);
  return 0;
}
```