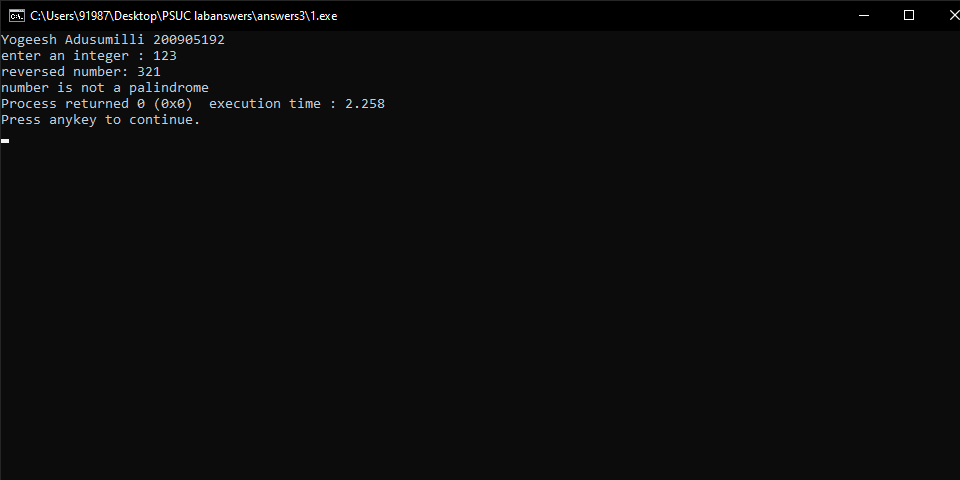
Lab no.3 – Looping Control Structures - While & Do-while loops

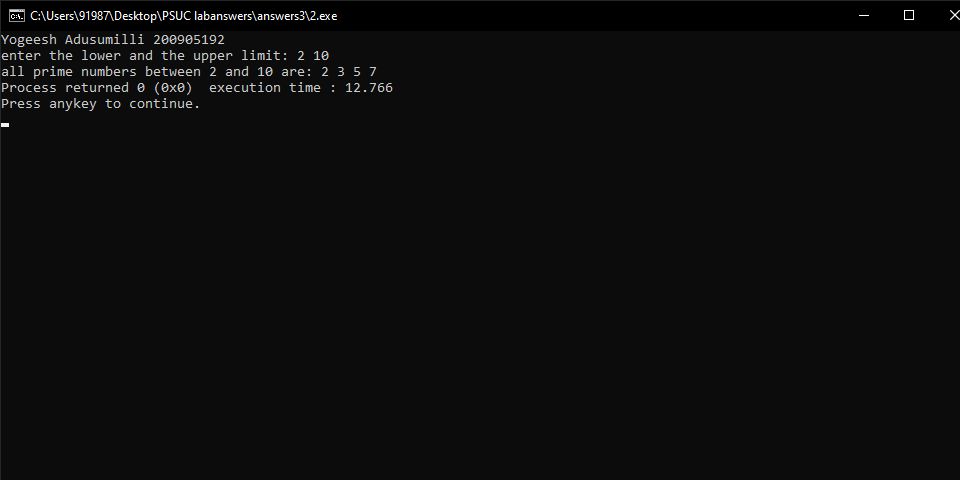
Q1) Reverse a given number and check if it is a palindrome or not. (use while loop).

#include <stdio.h>  
int main(){  
 printf("Yogeesh Adusumilli 200905192")  
 int num, last\_dig, rev\_num = 0;  
 printf("enter an integer : ");  
 scanf("%d", &num);  
 int num\_copy = num;  
 while (num)  
 {  
 last\_dig = num % 10;  
 rev\_num = rev\_num \* 10 + last\_dig;  
 num /= 10;  
 }  
 printf("reversed number: %d\n", rev\_num);  
 if (num\_copy == rev\_num)  
 {  
 printf("number is a palindrome");  
 }  
 else  
 {  
 printf("number is not a palindrome");  
 }  
 return 0;  
}



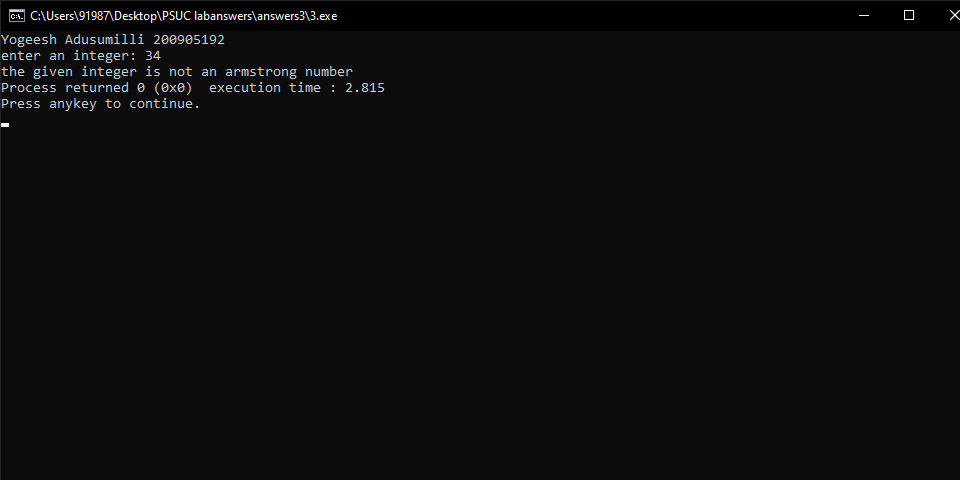
Q2) Generate prime numbers between 2 given limits.(use while loop).

#include <stdio.h>  
  
int isPrime(int n)  
{  
 for (int i=2; i <= n/2; i++)  
 {  
 if (n%i == 0)  
 return 0;  
 }  
 return 1;  
}  
  
int main(){  
 int upper\_limit, lower\_limit;  
 printf("Yogeesh Adusumilli 200905192")  
 printf("enter the lower and the upper limit: ");  
 scanf("%d %d", &lower\_limit, &upper\_limit);  
 int num = lower\_limit;  
 printf("all prime numbers between %d and %d are: ", lower\_limit, upper\_limit);  
 while (num != upper\_limit)  
 {  
 if (isPrime(num))  
 printf("%d ", num);  
 num += 1;  
 }  
 return 0;  
}



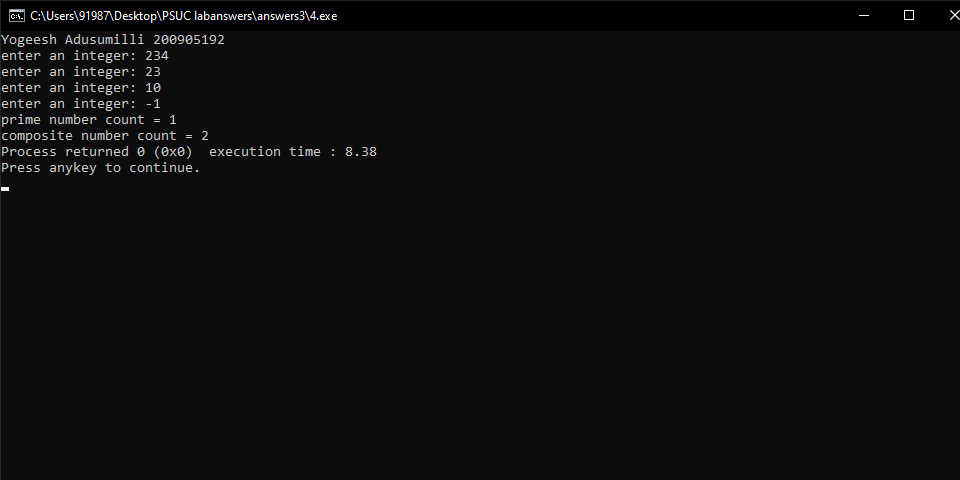
Q3) Check if the sum of the cubes of all digits of an inputted number equals the number itself (Armstrong Number). (use while loop)

#include <stdio.h>  
  
int main(){  
 printf("Yogeesh Adusumilli 200905192")  
 int num, temp\_num, last\_dig, res=0;  
 printf("enter an integer: ");  
 scanf("%d", &num);  
 temp\_num = num;  
 while (num)  
 {  
 last\_dig = num % 10;  
 res += last\_dig \* last\_dig \* last\_dig;  
 num /= 10;  
 }  
 if (temp\_num == res)  
 printf("the given integer is an armstrong number");  
 else  
 printf("the given integer is not an armstrong number");  
 return 0;  
}



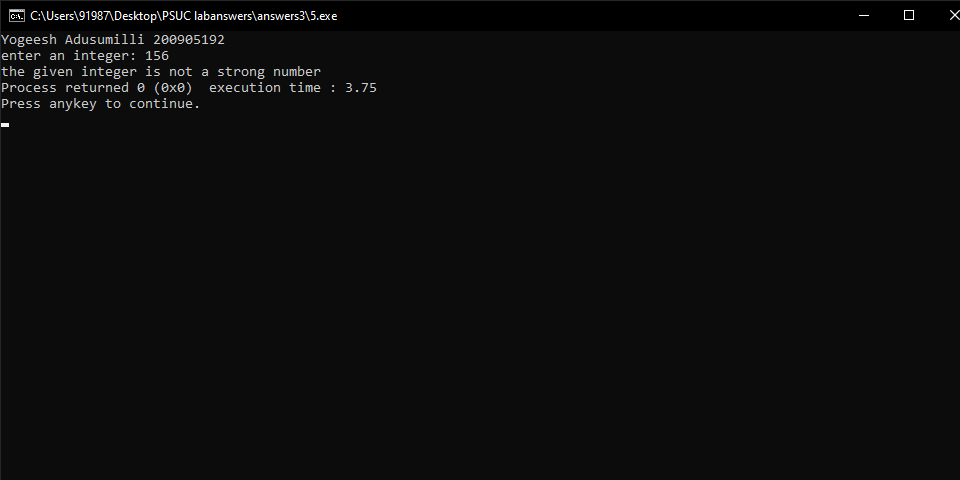
Q4) Write a program using do-while loop to read the numbers until -1 is encountered. Also count the number of prime numbers and composite numbers entered by user.

#include <stdio.h>  
  
int isPrime(int n)  
{  
 for (int i=2; i <= n/2; i++)  
 {  
 if (n%i == 0)  
 return 0;  
 }  
 return 1;  
}  
  
int main(){  
 printf("Yogeesh Adusumilli 200905192")  
 int num, prime\_count = 0, composite\_count = 0;  
 do  
 {  
 printf("enter an integer: ");  
 scanf("%d", &num);  
 if (num == 1 || num == -1)  
 continue;  
 if (isPrime(num))  
 prime\_count += 1;  
 else  
 composite\_count += 1;  
  
 } while(num != -1);  
  
 printf("prime number count = %d\n", prime\_count);  
 printf("composite number count = %d", composite\_count);  
 return 0;  
}



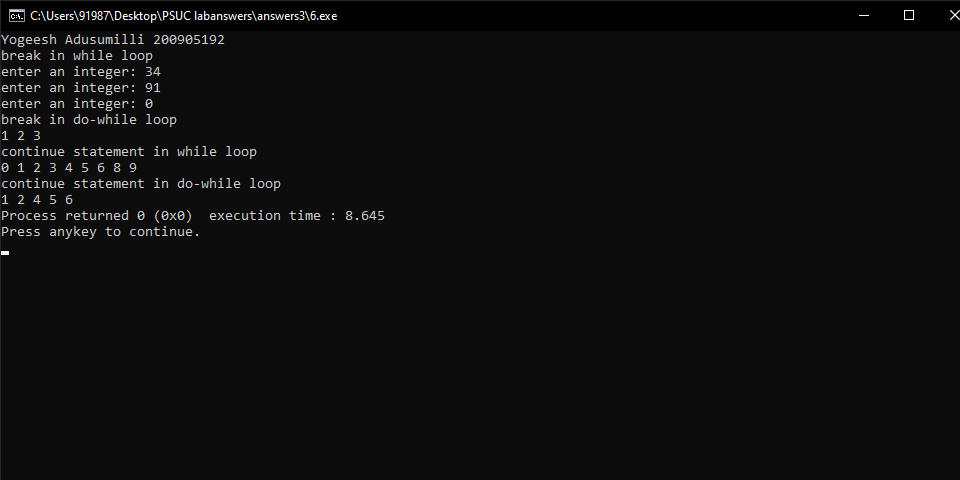
Q5) Check whether the given number is strong or not.

#include <stdio.h>  
  
int factorial(int n){  
 int res = 1;  
 while (n)  
 {  
 res \*= n;  
 n--;  
 }  
}  
  
int main(){  
 printf("Yogeesh Adusumilli 200905192")  
 int num, last\_dig, res = 0;  
 printf("enter an integer: ");  
 scanf("%d", &num);  
 int temp\_num = num;  
 while (num)  
 {  
 last\_dig = num % 10;  
 res += factorial(last\_dig);  
 num /= 10;  
 }  
 if (res == temp\_num)  
 printf("the given integer is a strong number");  
 else  
 printf("the given integer is not a strong number");  
 return 0;  
}



Q6) Write a program to demonstrate use of break and continue statements in while and do-while loops.

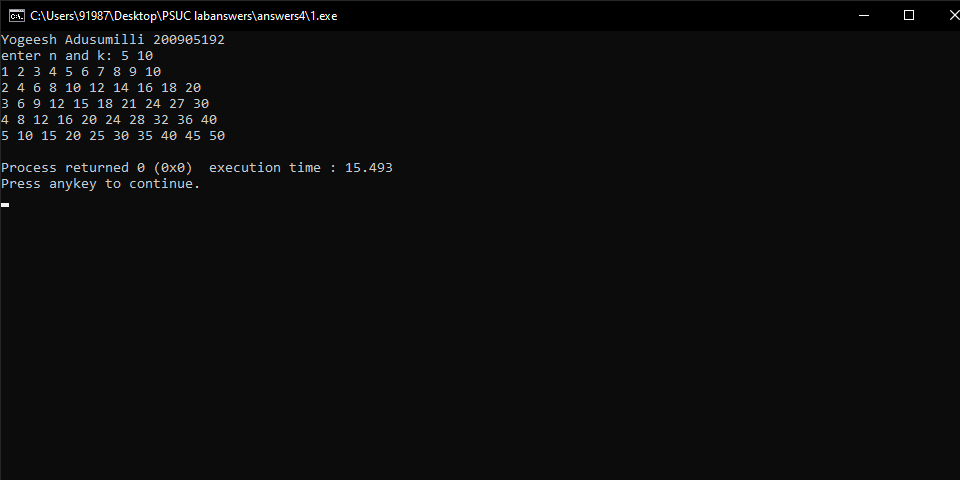
#include <stdio.h>  
  
int main(){  
 printf("Yogeesh Adusumilli 200905192")  
 int i = 1;  
 printf("break in while loop\n");  
 int n;  
 while (1)  
 {  
 printf("enter an integer: ");  
 scanf("%d", &n);  
 if (n == 0)  
 {  
 break;  
 }  
 }  
  
 printf("break in do-while loop\n");  
 i = 1;  
 do   
 {  
 if (i == 4)  
 {  
 break;  
 }  
 printf("%d ", i);  
 i++;  
 }while(i <= 10);  
  
 printf("\ncontinue statement in while loop\n");  
 i = 0;  
 while (i < 10)  
 {  
 if (i != 7)  
 {  
 printf("%d ", i);  
 i++;  
 }  
 else  
 {  
 i++;  
 continue;  
 }  
 }  
   
 printf("\ncontinue statement in do-while loop\n");  
 i = 1;  
 do  
 {  
 if (i==3)  
 {  
 i++;  
 continue;  
 }  
 printf("%d ", i);  
 i++;  
 }while(i<7);  
}



Lab no.4 – Looping Control Structures - For loops

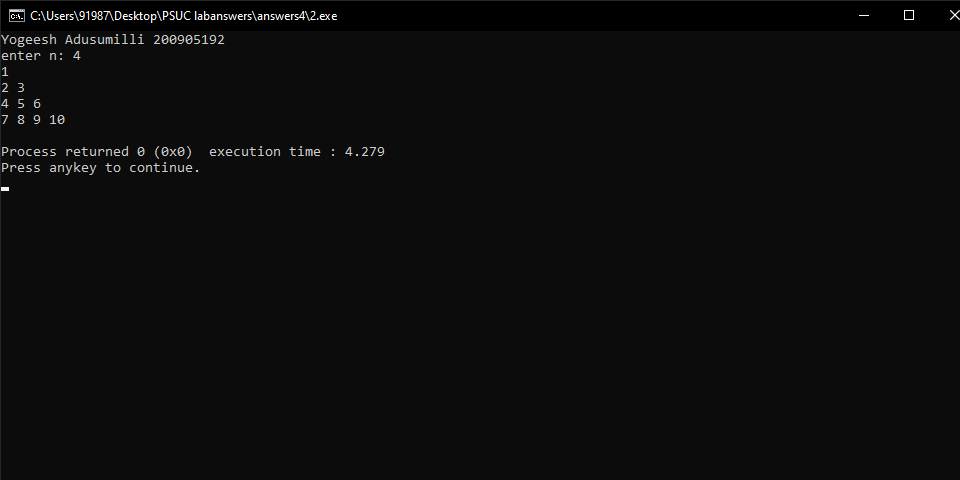
Q1) Generate the multiplication table for 'n' numbers up to 'k' terms (using nested for loops).

#include <stdio.h>  
  
int main(){  
 printf("Yogeesh Adusumilli 200905192")  
 int k, n;  
 printf("enter n and k: ");  
 scanf("%d %d", &n, &k);  
 for (int i = 1; i <= n; i++)  
 {  
 for (int j = 1; j <= k; j++)  
 {  
 printf("%d ", i \* j);  
 }  
 printf("\n");  
 }  
 return 0;  
}



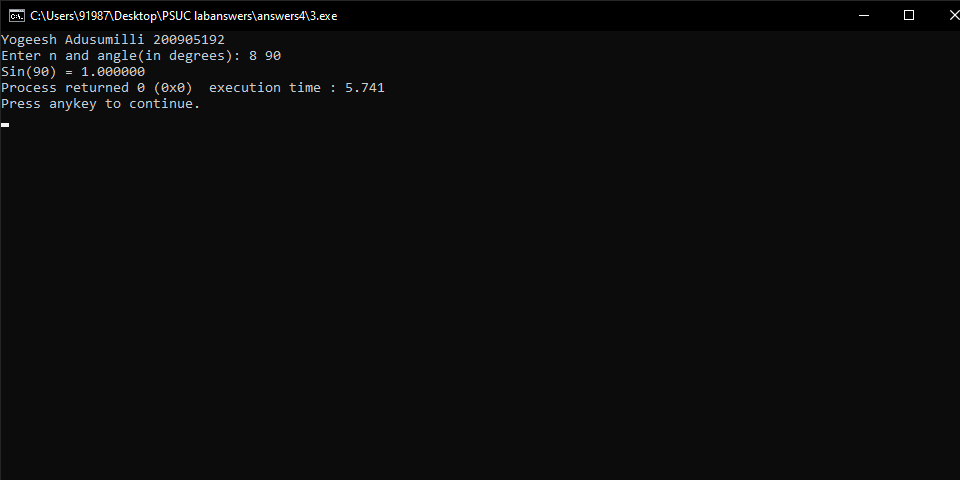
Q2) Generate Floyd's triangle using natural numbers for a given limit N. (using for loops).

#include <stdio.h>  
  
int main(){  
 printf("Yogeesh Adusumilli 200905192")  
 int n, current\_number = 1;  
 printf("enter n: ");  
 scanf("%d", &n);  
 for (int i = 1; i <= n; i++)  
 {  
 for (int j = 1; j <= i; j++)  
 {  
 printf("%d ", current\_number);  
 current\_number++;  
 }  
 printf("\n");  
 }  
 return 0;  
}



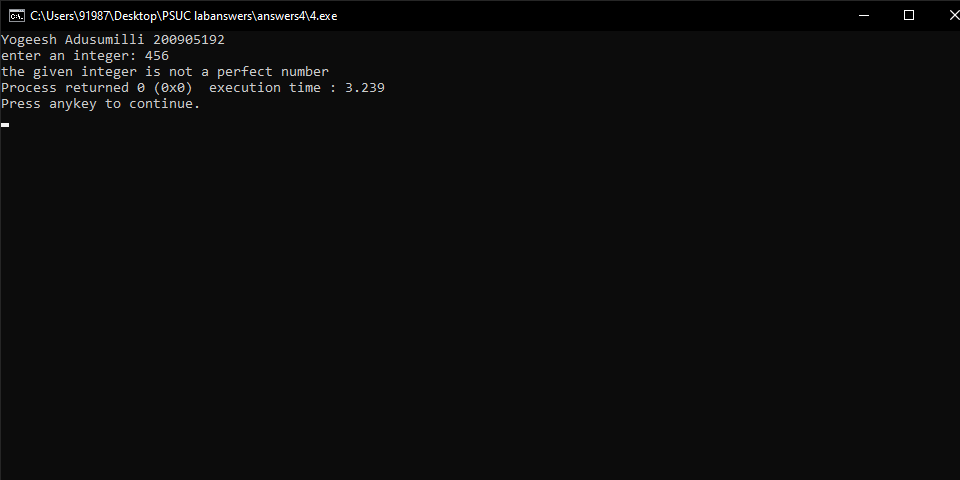
Q3) Evaluate the sine series, sin(x)= x- x3/3! + x5/5!â€“x7/7!+ .... to n terms.

#include<stdio.h>  
#include<math.h>  
  
int factorial(int n){  
 int fact = 1;  
 for (int i = 1; i <= n; i++)  
 {  
 fact \*= i;  
 }  
 return fact;  
}  
  
int main()  
{  
 printf("Yogeesh Adusumilli 200905192")  
   
 int degrees, n, sign = 1;  
 printf("Enter n and angle(in degrees): ");  
 scanf("%d %d", &n, &degrees);  
   
 float radians = degrees \* (3.14159 / 180.0);  
 float sum = 0;  
  
 for (int power = 1; power <= n \* 2; power += 2)  
 {  
 sum += sign \* (pow(radians, power) / factorial(power));  
 sign \*= -1;  
 }  
 printf("Sin(%d) = %f", degrees, sum);  
 return 0;  
}



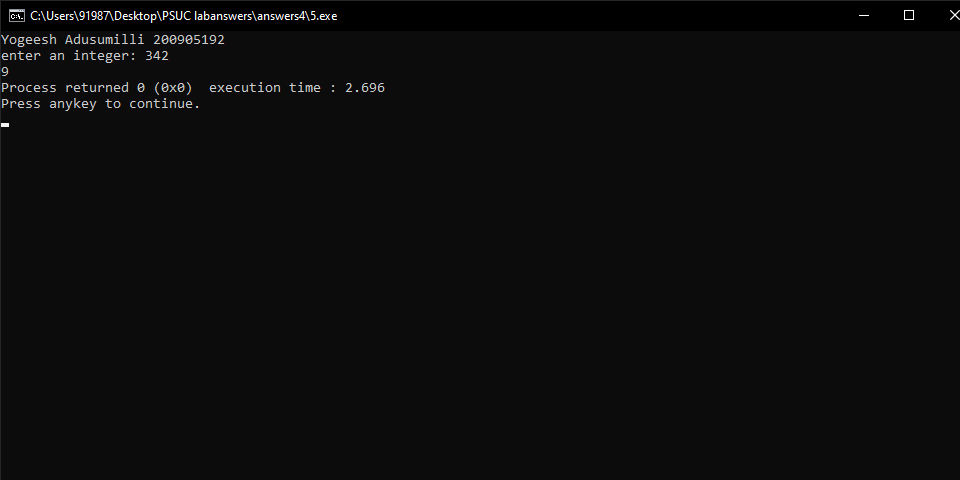
Q4) Check whether a given number is perfect or not

#include <stdio.h>  
  
int getFactorSum(int n)  
{  
 int factorSum = 0;  
 for (int i = 1; i <= n-1; i++)  
 {  
 if (n % i == 0)  
 factorSum += i;  
 }  
 return factorSum;  
}  
  
  
int main(){  
 printf("Yogeesh Adusumilli 200905192")  
 int n;  
 printf("enter an integer: ");  
 scanf("%d", &n);  
 if (n == getFactorSum(n))  
 printf("the given integer is a perfect number");  
 else  
 printf("the given integer is not a perfect number");  
 return 0;  
}



Q5) Find out the generic root of any number.

#include <stdio.h>  
  
int main(){  
 printf("Yogeesh Adusumilli 200905192")  
 int n, temp, last\_dig, sum = 0;  
 printf("enter an integer: ");  
 scanf("%d", &n);  
 temp = n;  
 for (;;)  
 {  
 for (;;)  
 {  
 if (temp == 0)  
 break;  
 last\_dig = temp % 10;  
 sum += last\_dig;  
 temp /= 10;  
 }  
 if (sum <= 9)  
 break;  
 else  
 {  
 temp = sum;  
 sum = 0;  
 }  
 }  
 printf("%d ", sum);  
 return 0;  
}



Q6) Write a program to demonstrate use of break and continue statements in for loop.

#include <stdio.h>  
  
int main()  
{  
 printf("Yogeesh Adusumilli 200905192")  
 int n, flag = 0;  
 printf("break statement in for loop\n");  
 printf("enter n: ");  
 scanf("%d", &n);  
 for (int i = 2; i <= n/2; i++)  
 {  
 if (n % i == 0)  
 {  
 flag = 1;  
 break;  
 }   
 }  
 if (flag == 0)  
 printf("%d is a prime number.",n);  
 else  
 printf("%d is not a prime number.",n);  
  
 printf("\ncontinue statement in for loop\n");  
 for (int i = 0; i < 8; i++)  
 {  
 if (i == 5)  
 {  
 continue;  
 }  
 printf("%d ", i);  
 }  
}

