*DAY 1 PROGRAMS*

*NAME: PAMIDI SAIVENKAT*

*REG.NO:192111197*

1. TIME COMPLEXITY USING COUNTER METHOD

CODE:

#include <stdio.h>

void function(int min);

int main()

{

int n;

scanf("%d",&n);

function(n);

return 0;

}

void function(int n)

{

int count=0;

int i=1,s=1;

count++;

count++;

while(s<=n)

{

count++;

i++;

count++;

s+=i;

count++;

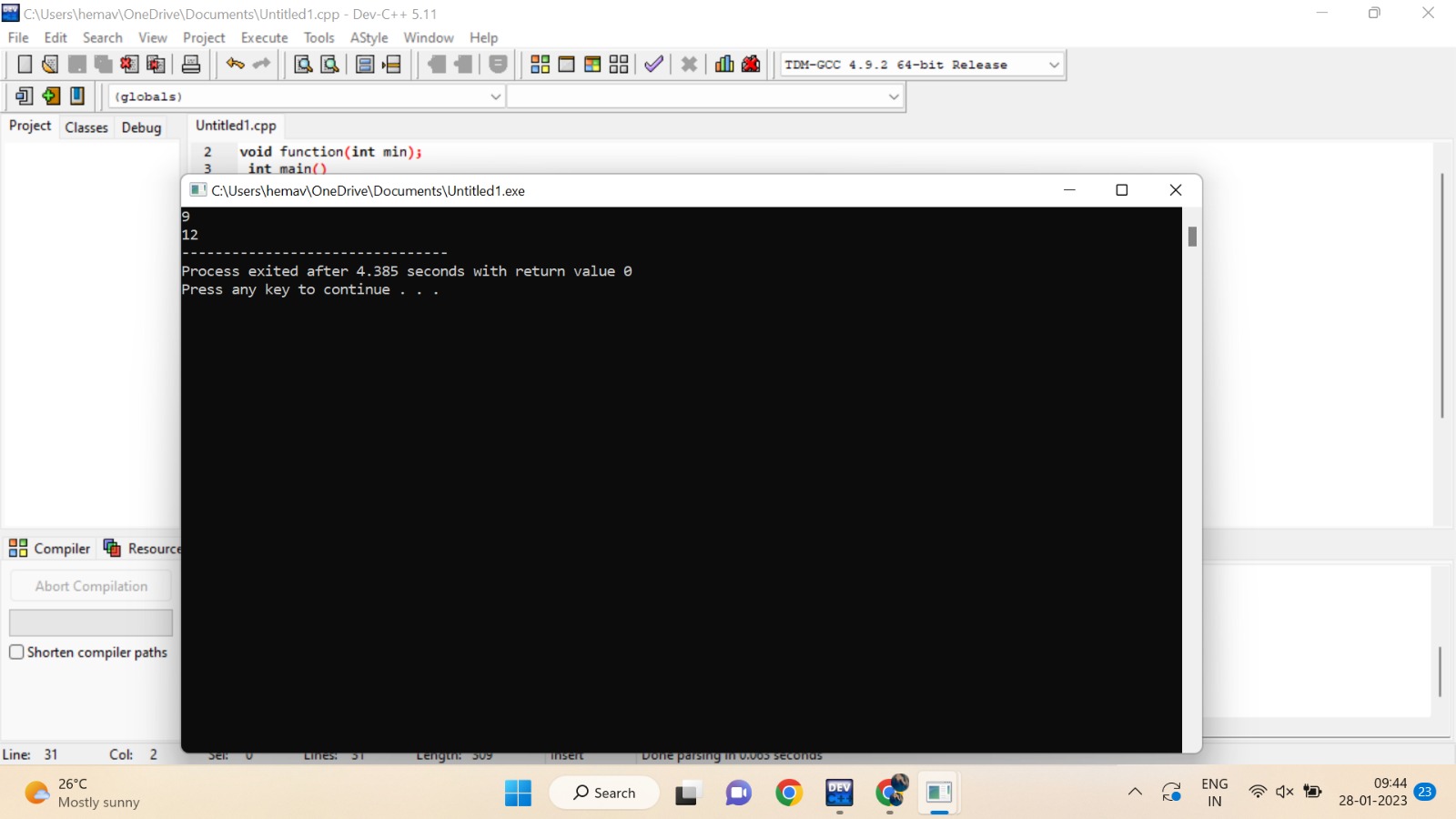
}

count++;

printf("%d",count);

}

OUTPUT:



1. TIME COMPLEXITY USING COUNTER METHOD

CODE:

#include <stdio.h>

void function(int n);

int main()

{

int n;

scanf("%d",&n);

function(n);

return 0;

}

void function(int n)

{

int count=0;

if(n==1)

{

count++;

count++;

}

else

{

count++;

for(int i=1;i<=n;i++)

{

count++;

for(int j=1;j<=n;j++)

{

count++; count++; count++; count++; break;

}

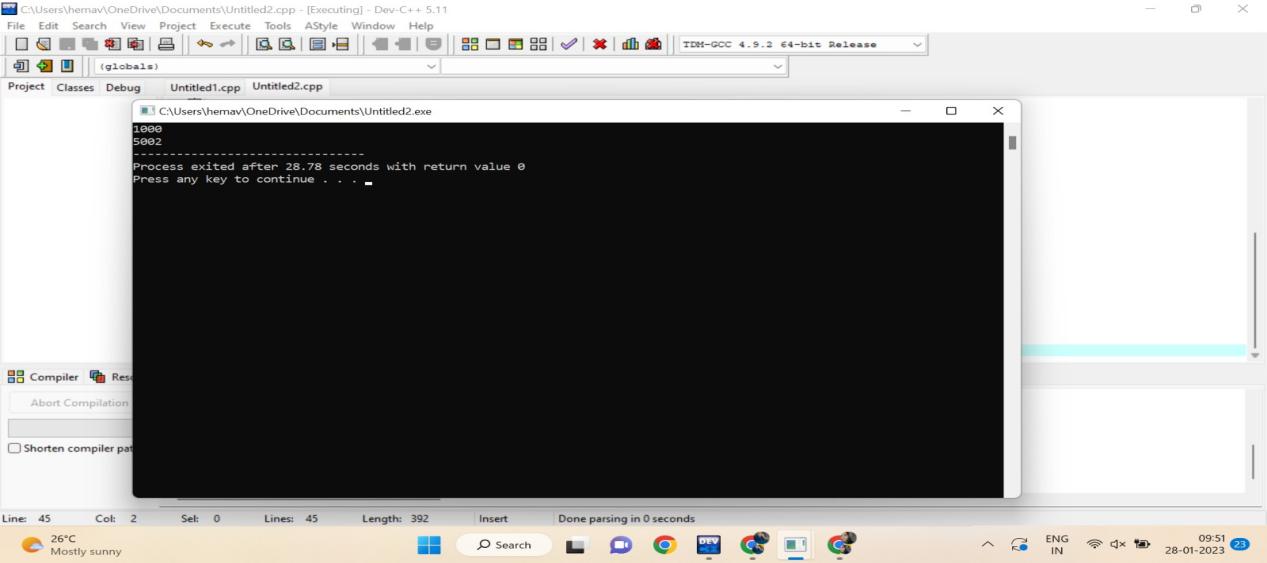
}

count++;

}

printf("%d",count);

}



1. TIME COMPLEXITY USING COUNTER METHOD

CODE:

#include <stdio.h>

int factor(int n);

int count=0;

int main()

{

int n;

scanf("%d",&n);

factor(n);

printf("%d",count);

return 0;

}

int factor(int n)

{

int i; count++;

for(i=1;i<=n;++i)

{

count++;

if(n%i==0)

{

//print

}

count++;

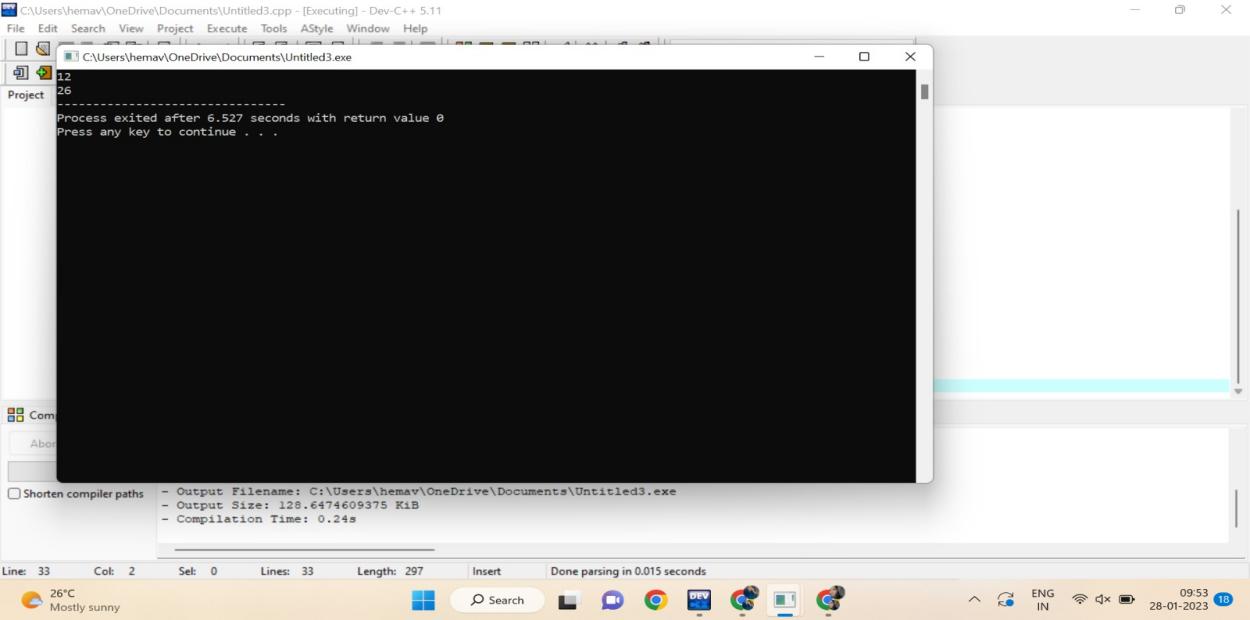
}

count++;

return 0;

}

OUTPUT:



1. TIME COMPLEXITY USING COUNTER METHOD

CODE:

#include <stdio.h>

void function(int n);

int main()

{

int n;

scanf("%d",&n);

function(n);

return 0;

}

void function(int n)

{

int count=0;

int c=0;

count++;

for(int i=n/2;i<n;i++)

{

count++;

for(int j=1;j<n;j=2\*j)

{

count++;

for(int k=1;k<n;k=k\*2)

{

count++; c++;

count++;

}

count++;

}

count++;

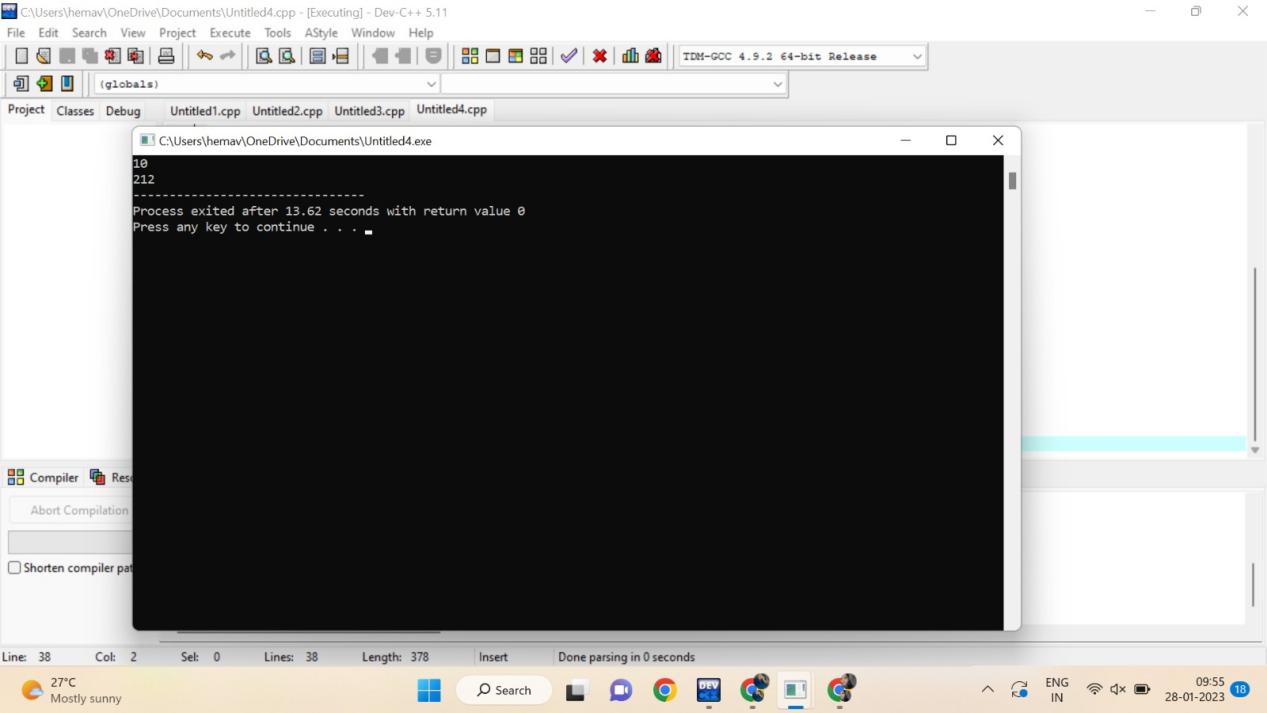
}

count++;

printf("%d",count);

}

OUTPUT:



1. TIME COMPLEXITY USING COUNTER METHOD

CODE:

#include <stdio.h>

void reverse(int n);

int main()

{

int n; scanf("%d",&n);

reverse(n);

return 0;

}

void reverse(int n)

{

int count=0;

int rev=0,remainder;

count++;

while(n!=0)

{

count++;

remainder=n%10;

count++;

rev=rev\*10+remainder;

count++;

n=n/10; count++;

}

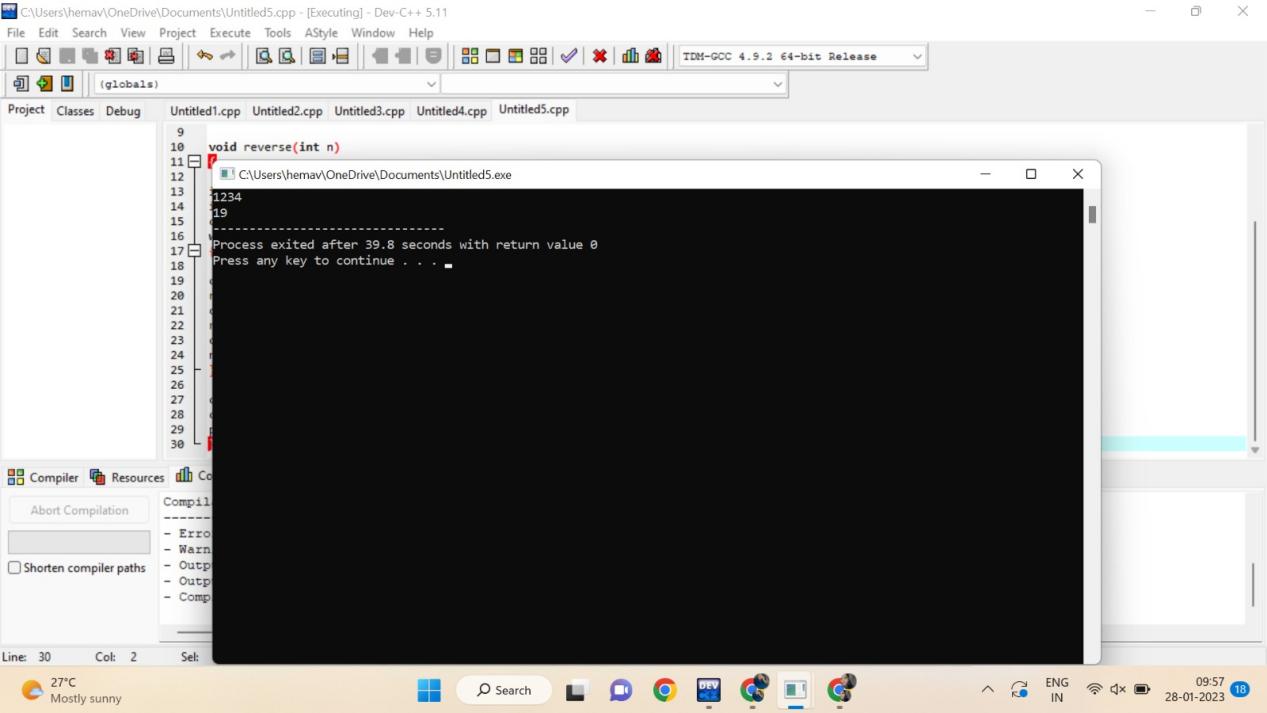
count++;

count++;

printf("%d",count);

}

OUTPUT:



1. CHECK A NUMBER IS ARMSTRONG NUMBER OR NOT

CODE:

#include<stdio.h>

int main()

{

int n,r,sum=0,temp;

int count=0;

printf("enter the number=");

scanf("%d",&n);

temp=n;

while(n>0)

{

count++;

r=n%10;

count++;

sum=sum+(r\*r\*r);

count++;

n=n/10;

count++;

}

count++;

if(temp==sum)

printf("armstrong number ");

else

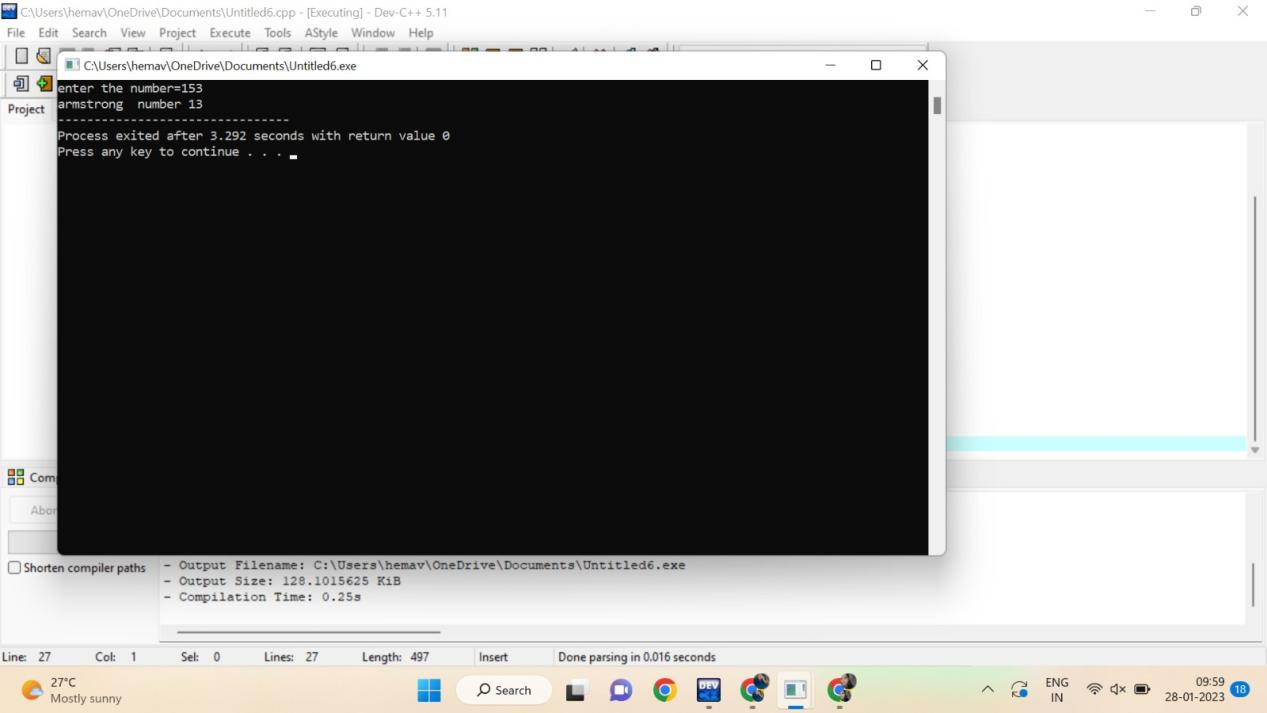
printf("not armstrong number");

printf("%d",count);

return 0;

}

OUTPUT:



1. CHECK A NUMBER IS PRIME NUMBER OR NOT

CODE:

#include <stdio.h>

int main() {

int n, i, flag = 0;

int count=0;

printf("Enter a positive integer: ");

scanf("%d", &n);

count++;

if (n == 0 || n == 1)

flag = 1;

count++;

for (i = 2; i <= n / 2; ++i) {

count++;

if (n % i == 0) {

flag = 1;

break;

count++;

}

}

count++;

if (flag == 0)

printf("%d is a prime number.", n);

else

printf("%d is not a prime number.", n);

printf("%d",count);

return 0;

}

OUTPUT:

