“SESSION – 4”

Q] Write a C function to calculate the Power of a Number.

ALGORITHM

STEP 1: ***Start***

STEP 2: ***Declare the Power Function.***

STEP 3: ***Initialize integer value of result to 1.***

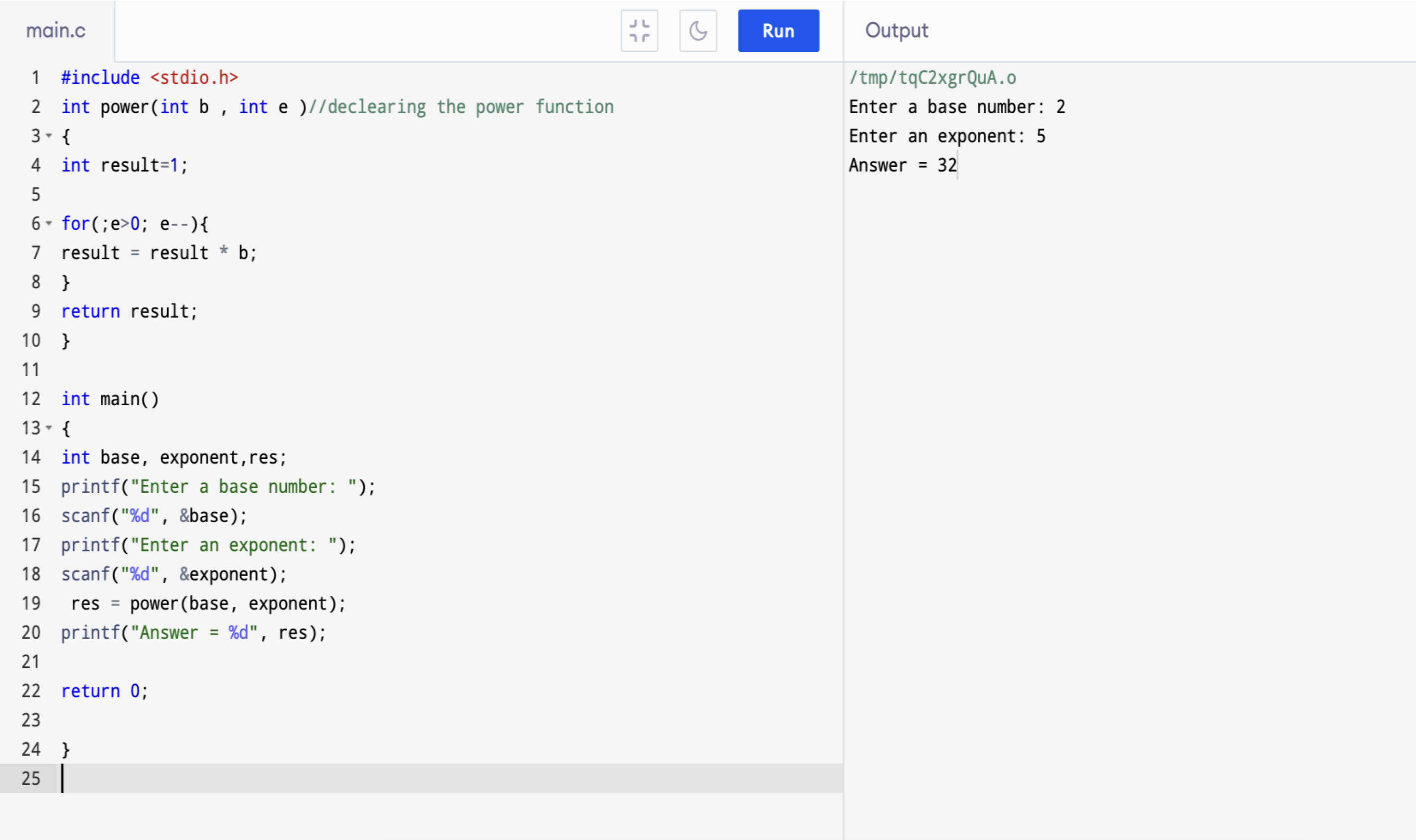
STEP 4: ***Run the For Loop(;e>0;e--) and print result=*b and return result.***

STEP 5: ***Under the Main Function, initialize integer values of base, exponent and res.***

STEP 6: ***Print enter a base number and enter an exponent.***

STEP 7: ***res=power(base, exponent)and now print Answer is ,res.***

STEP 8: ***Stop.***



```c
#include <stdio.h>
int power(int b , int e )//declearing the power function
{
int result=1;

for(;e>0; e--){
result = result * b;
}
return result;
}

int main()
{
int base, exponent,res;
printf("Enter a base number: ");
scanf("%d", &base);
printf("Enter an exponent: ");
scanf("%d", &exponent);
 res = power(base, exponent);
printf("Answer = %d", res);

return 0;

}

```

```
/tmp/tqC2xgrQuA.o
Enter a base number: 2
Enter an exponent: 5
Answer = 32
```

Q] Write a C function to calculate the Binary equivalent of a Decimal Number. *(Note: Validate your program for a) (53)10).*

ALGORITHM

STEP 1: ***Start.***

STEP 2: ***Declare the Decimal to Binary Function .***

STEP 3: ***Take integer values of an array a and initialize it to 10 and also take integer values of n,i.***

STEP 4: ***Print the statement Enter the Decimal number.***

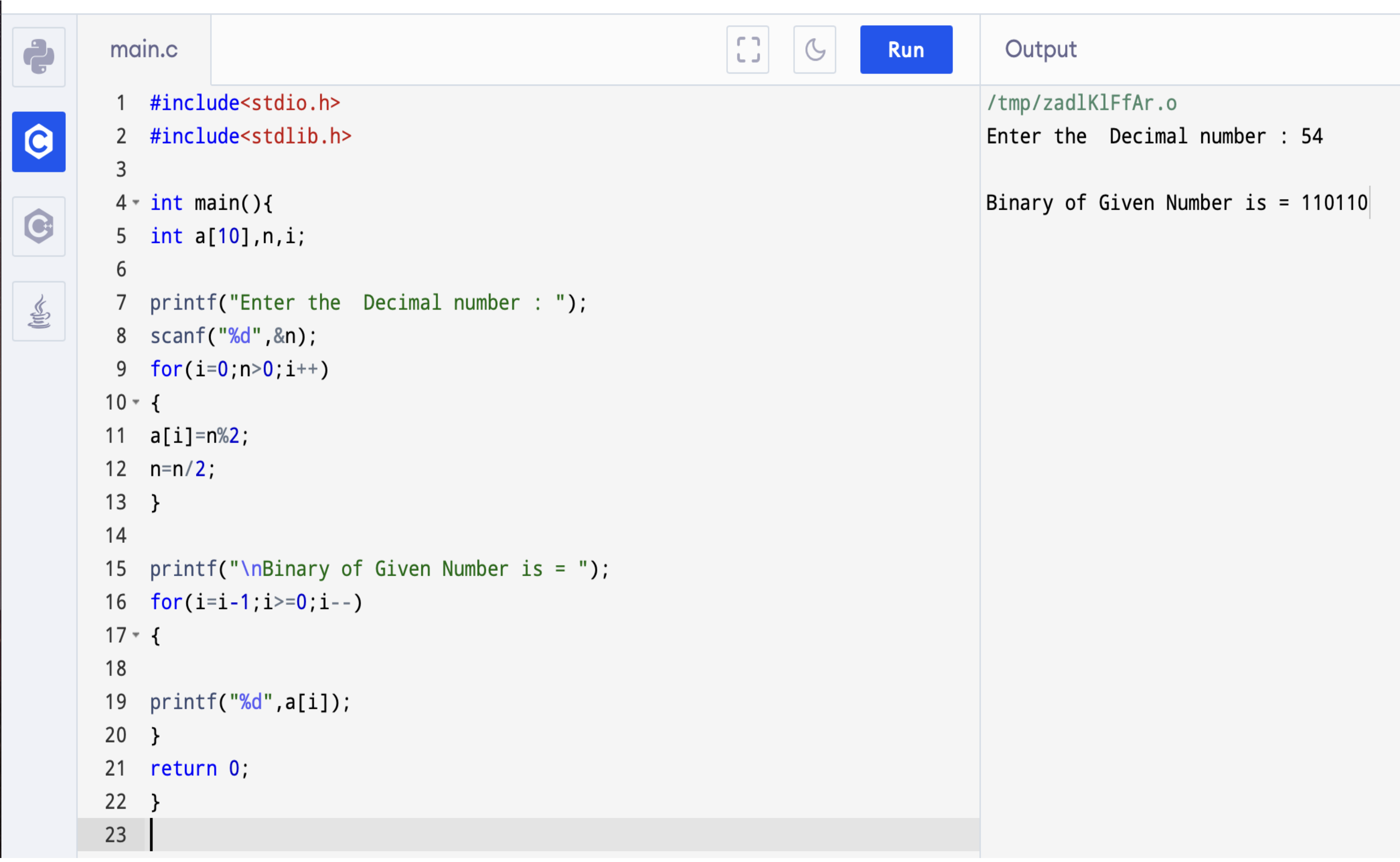
STEP 5: ***Run a For Loop i=0;n>0;i++.***

STEP 6: ***Put a[i]=n%2 and initialize n=n/2.***

STEP 7: ***Print the statement Binary of Given number is.***

STEP 8: ***Again run a For Loop i=i-1;i>=0;i—and print a[i] orelse come out of the loop and return 0.***

STEP 9: ***Stop.***



```c
#include<stdio.h>
#include<stdlib.h>

int main(){
int a[10],n,i;

printf("Enter the  Decimal number : ");
scanf("%d",&n);
for(i=0;n>0;i++)
{
a[i]=n%2;
n=n/2;
}

printf("\nBinary of Given Number is = ");
for(i=i-1;i>=0;i--)
{

printf("%d",a[i]);
}
return 0;
}

```

```
/tmp/zadlKlFfAr.o
Enter the  Decimal number : 54

Binary of Given Number is = 110110
```

Q] Write a recursive C function to print the Fibonacci Series.

ALGORITHM

STEP 1: ***Start***

STEP 2: ***Declare the f Function .***

STEP 3: ***In the Main Function, initialize integer values of n and i and also initialize integer value of m to 0.***

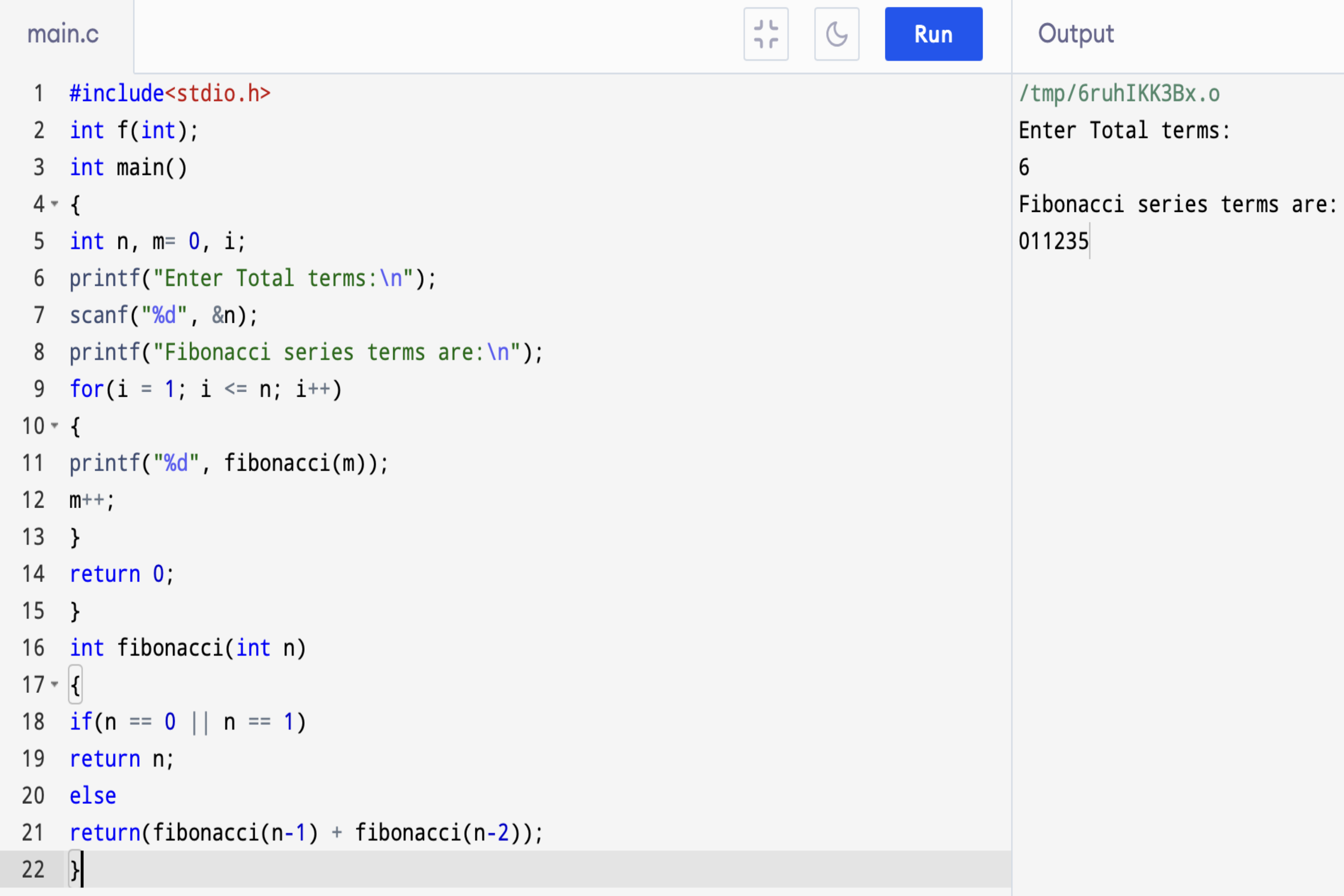
STEP 4: ***Run the For Loop(i=1;i<=n;i++)***

STEP 5: ***Print Fibonacci(m) and increment m by 1.***

STEP 6: ***int Fibonacci(int n) and run a if-else statement.***

STEP 7: ***If (n= = 0 || n = = 1) then return n.***

STEP 8: ***Else return (fibonacci(n-1)+fibonacci(n-2))***
STEP 9: ***Stop.***



```c
#include<stdio.h>
int f(int);
int main()
{
int n, m= 0, i;
printf("Enter Total terms:\n");
scanf("%d", &n);
printf("Fibonacci series terms are:\n");
for(i = 1; i <= n; i++)
{
printf("%d", fibonacci(m));
m++;
}
return 0;
}
int fibonacci(int n)
{
if(n == 0 || n == 1)
return n;
else
return(fibonacci(n-1) + fibonacci(n-2));
}
```

```
/tmp/6ruhIKK3Bx.o
Enter Total terms:
6
Fibonacci series terms are:
011235
```

Q] Write a recursive C function to calculate Factorial of an Integer.

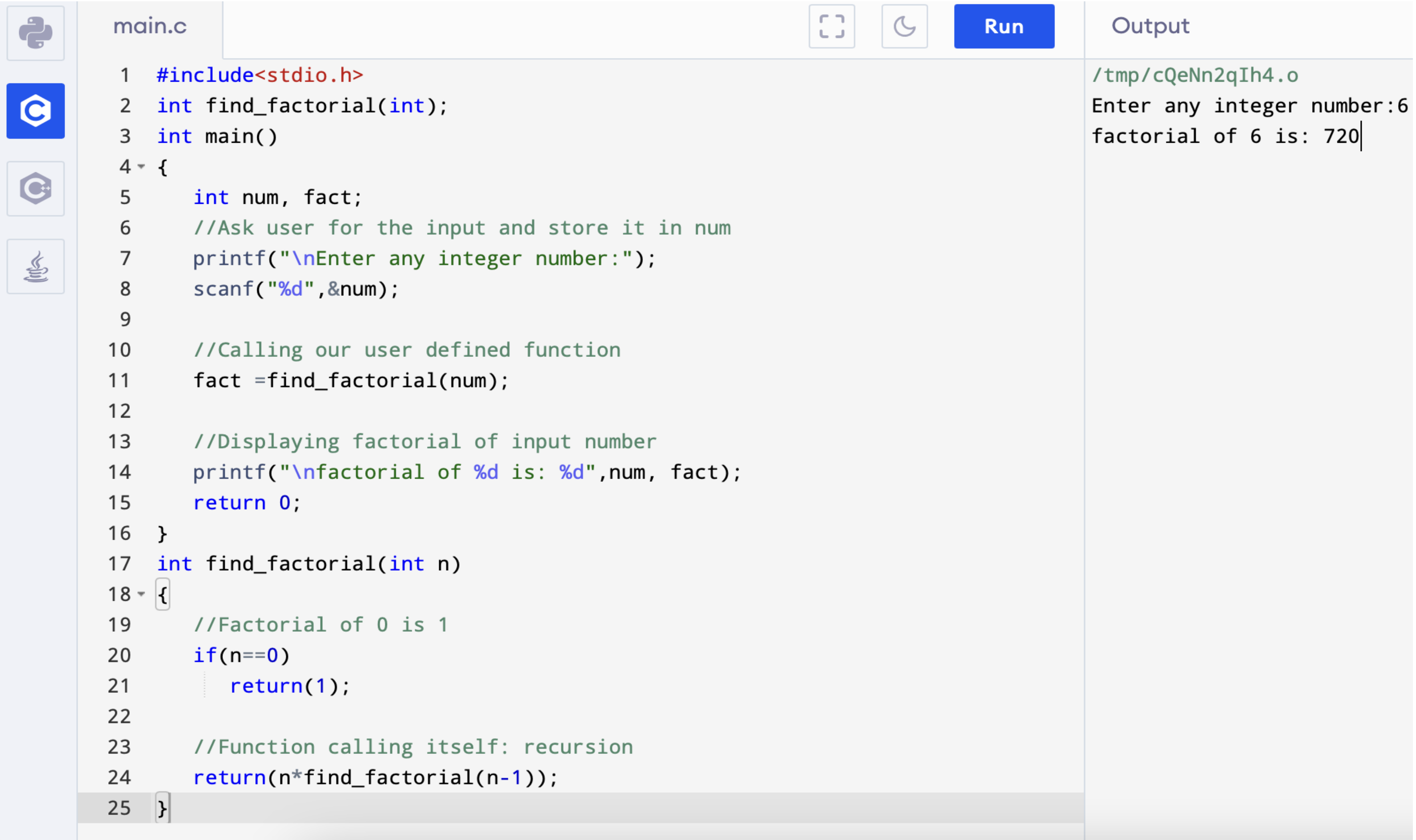
ALGORITHM

STEP 1:   ***Start***
STEP 2: ***Declare the find_factorial Function .***
STEP 3: ***In the Main Function, initialize integer values of num and fact.***
STEP 4: ***Print the Statement Enter Any Integer Number.***
STEP 5: ***Call the User defined Function ie, fact=find_factorial(num).***
STEP 6: ***Print The Statement Factorial of .***

***STEP 7:* Factorial of 0 is 1**

***STEP 8:* Function calling itself : recursion, n* find_factorial(n-1).**

***STEP 9:* Stop.**



```c
#include<stdio.h>
int find_factorial(int);
int main()
{
   int num, fact;
   //Ask user for the input and store it in num
   printf("\nEnter any integer number:");
   scanf("%d",&num);

   //Calling our user defined function
   fact =find_factorial(num);

   //Displaying factorial of input number
   printf("\nfactorial of %d is: %d",num, fact);
   return 0;
}
int find_factorial(int n)
{
   //Factorial of 0 is 1
   if(n==0)
      return(1);

   //Function calling itself: recursion
   return(n*find_factorial(n-1));
}
```

```
/tmp/cQeNn2qIh4.o
Enter any integer number:6
factorial of 6 is: 720
```

Q] Write a recursive C function to calculate the Sum of All Digits of a Given Integer.

ALGORITHM

***STEP 1:* Start**

***STEP 2:* Declare the sum Function .**

***STEP 3:* In the Main Function, initialize integer values of num and result.**

***STEP 4:* Print the Statement Enter the number.**

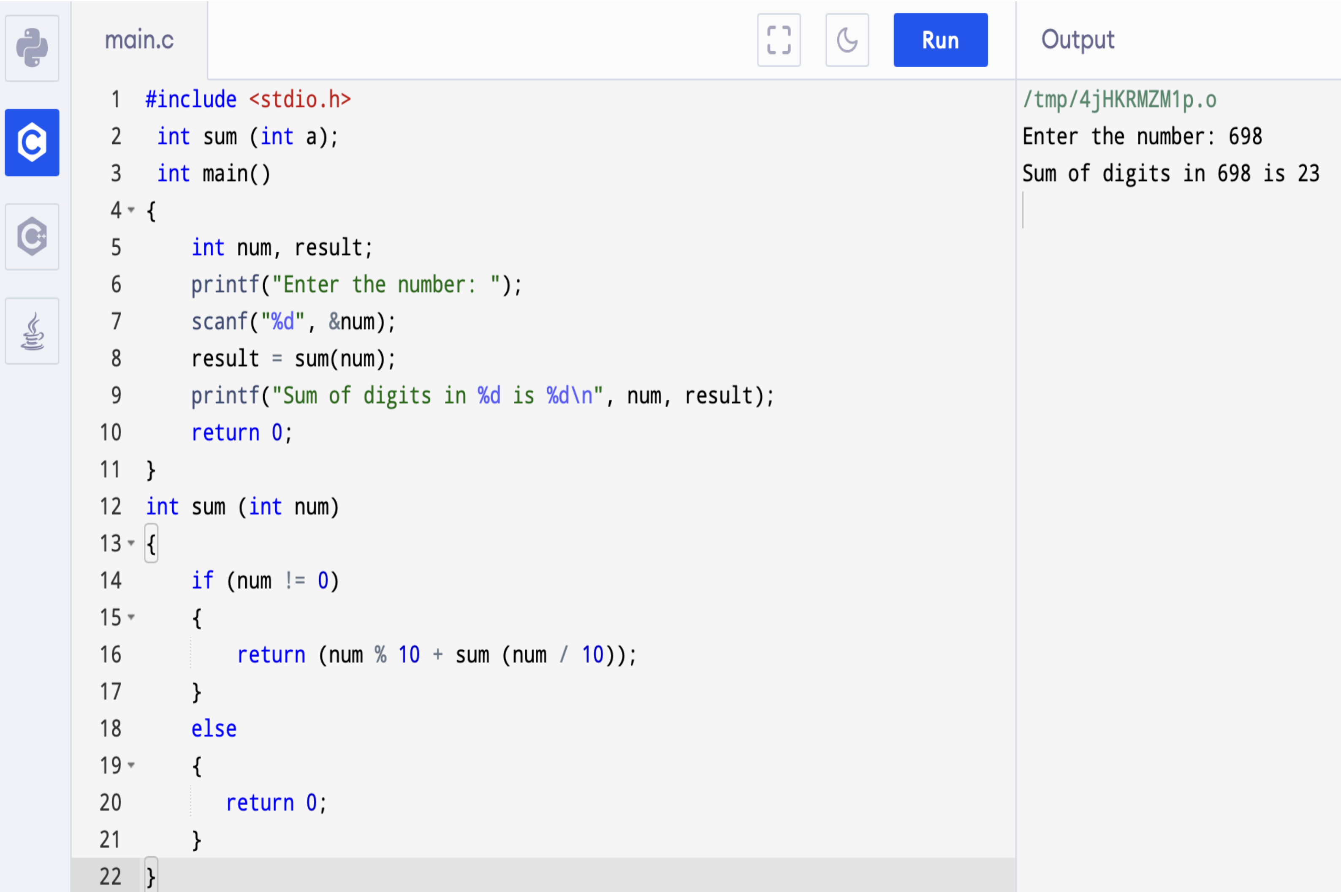
***STEP 5:* result = sum(num).**

***STEP 6:* Print The Statement Sum of Digits in.**

STEP 7: ***If (num ! =0), return (num%10+sum(num/10))***

STEP 8: ***Else return 0.***

STEP 9: ***Stop.***



```c
#include <stdio.h>
 int sum (int a);
 int main()
{
    int num, result;
    printf("Enter the number: ");
    scanf("%d", &num);
    result = sum(num);
    printf("Sum of digits in %d is %d\n", num, result);
    return 0;
}
int sum (int num)
{
    if (num != 0)
    {
        return (num % 10 + sum (num / 10));
    }
    else
    {
       return 0;
    }
}
```

```
/tmp/4jHKRMZM1p.o
Enter the number: 698
Sum of digits in 698 is 23
```

THANK YOU