KIDUS ABEBE MEKONEN
Roll No- 2106285
Section A11
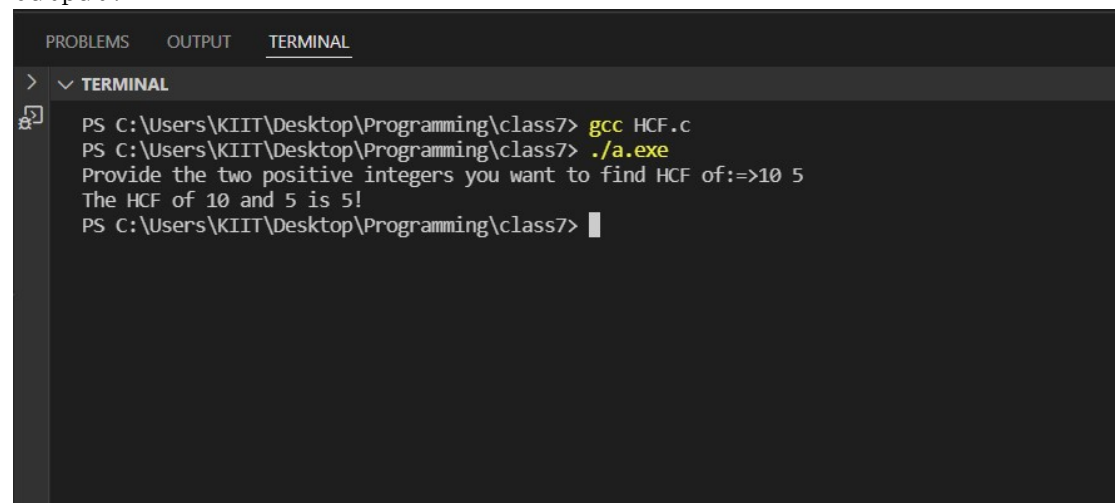**IT department**

Class 7 HOME questions

#1. WAP to find the GCD/HCF of two numbers .
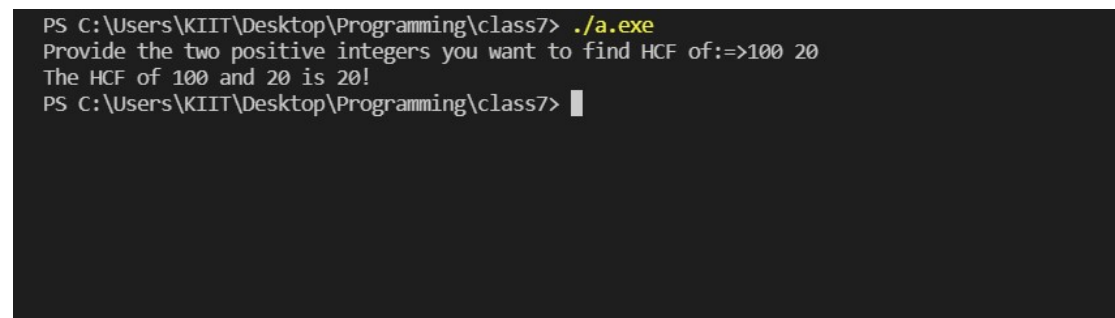
Code:

```c
#include <stdio.h>
#include <conio.h>
int main()
{
    int a285,b285,i285,mod285;
    printf("Provide the two positive integers you want to find HCF of:=>");
    scanf("%d%d",&a285,&b285);
    for(i285=1;i285<=a285 && i285<=b285;i285++)
    {
        if(a285%i285==0 && b285%i285==0)
            {
                mod285=i285;
            }
    }
    printf("The HCF of %d and %d is %d!",a285,b285,mod285);
    return 0;
}
```

Output:

```
PROBLEMS    OUTPUT    TERMINAL

> ∨ TERMINAL

  PS C:\Users\KIIT\Desktop\Programming\class7> gcc HCF.c
  PS C:\Users\KIIT\Desktop\Programming\class7> ./a.exe
  Provide the two positive integers you want to find HCF of:=>10 5
  The HCF of 10 and 5 is 5!
  PS C:\Users\KIIT\Desktop\Programming\class7>
```

```
PS C:\Users\KIIT\Desktop\Programming\class7> ./a.exe
Provide the two positive integers you want to find HCF of:=>100 20
The HCF of 100 and 20 is 20!
PS C:\Users\KIIT\Desktop\Programming\class7>
```

#2. WAP to check whether a number n is prime number or not.

Code:

```c
#include <stdio.h>
int main()
{
    int a285,j,i,b,count;
```
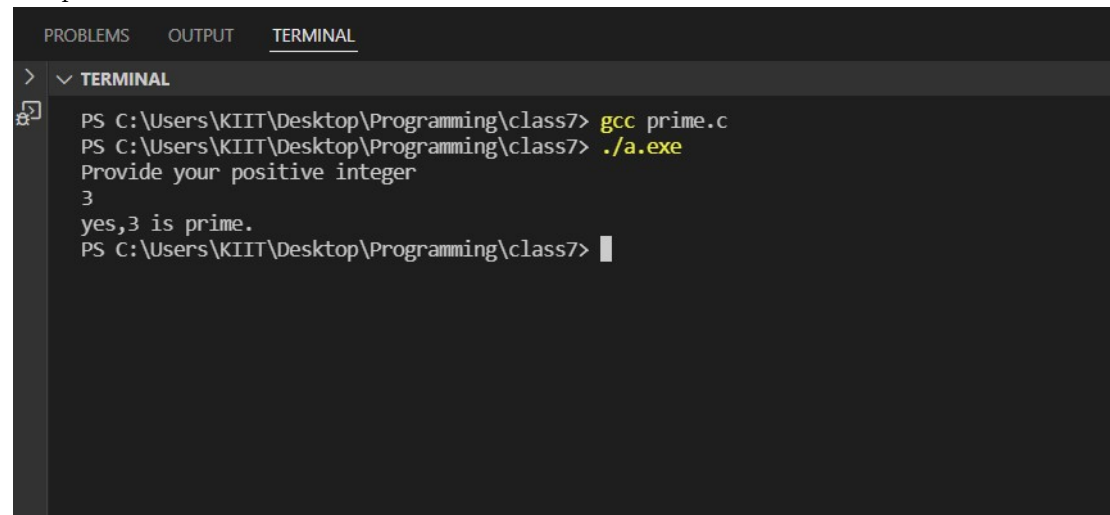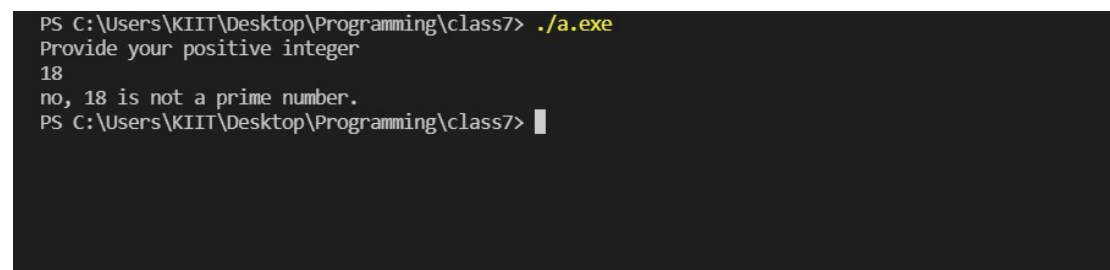
```c
    printf("Provide your positive integer\n");
    scanf("%d",&a285);
    count = 0;
    if(a285==0 || a285==1)
        count=1;
    for(i=2;i<a285;i++)
    {
    if (a285%i==0)
    count++;
    }
    if(count==0)
    {
        printf("yes,%d is prime.",a285);
    }
    else
        printf("no, %d is not a prime number.", a285);
    return 0;
}
```

Output:

#3. WAP to check whether an input integer is perfect number or not.
(Perfect number is a number which is equal to sum of its divisor. For
example, divisors of 6 are 1, 2 and 3. The sum of these divisors is
6.)
Code:

```c
#include <stdio.h>
int main()
{
    int a285,i285,count285=0;
```
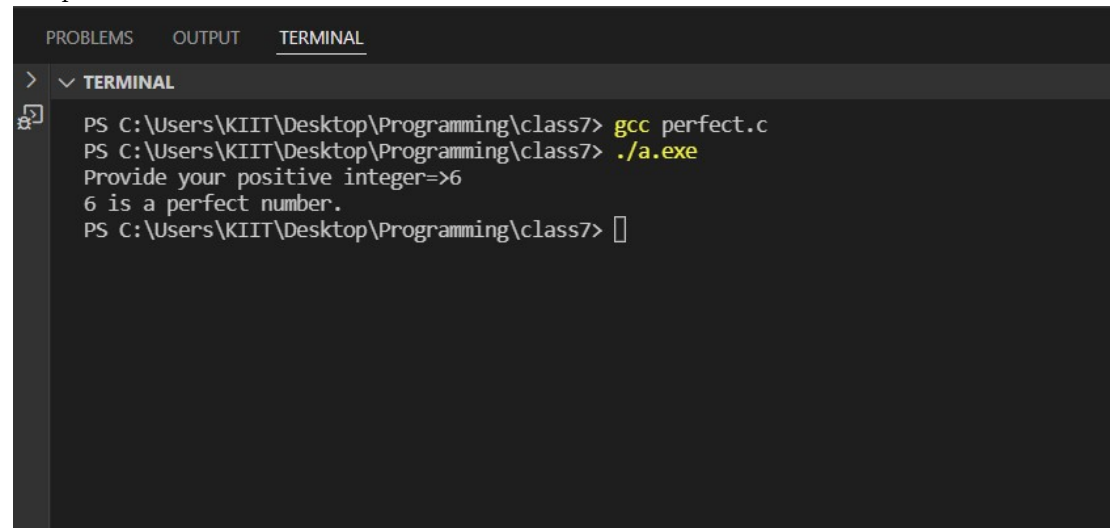
```c
    printf("Provide your positive integer=>");
    scanf("%d",&a285);
    count285=0;
    for(i285=1;i285<a285;i285++)
    {
        if((a285%i285)==0)
            {
                count285+=i285;
            }
    }
    if(count285==a285)
    printf("%d is a perfect number.\n",a285);
    else
    printf("%d is not a perfect number.\n",a285);
    return 0;
}
```

Output:

PROBLEMS   OUTPUT   TERMINAL

> ∨ TERMINAL

```
PS C:\Users\KIIT\Desktop\Programming\class7> gcc perfect.c
PS C:\Users\KIIT\Desktop\Programming\class7> ./a.exe
Provide your positive integer=>6
6 is a perfect number.
PS C:\Users\KIIT\Desktop\Programming\class7> []
```

```
PS C:\Users\KIIT\Desktop\Programming\class7> ./a.exe
Provide your positive integer=>80
80 is not a perfect number.
PS C:\Users\KIIT\Desktop\Programming\class7> ▊
```

#4. WAP to print the series as 1 2 7 15 31 ....n, where n is given by user.

Code:

```c
#include<stdio.h>
int main()
{
    int n285,i285,pr285=0;
    printf("Enter the range of number:");
    scanf("%d",&n285);
    for(i285=1;i285<=n285;i285++)
```
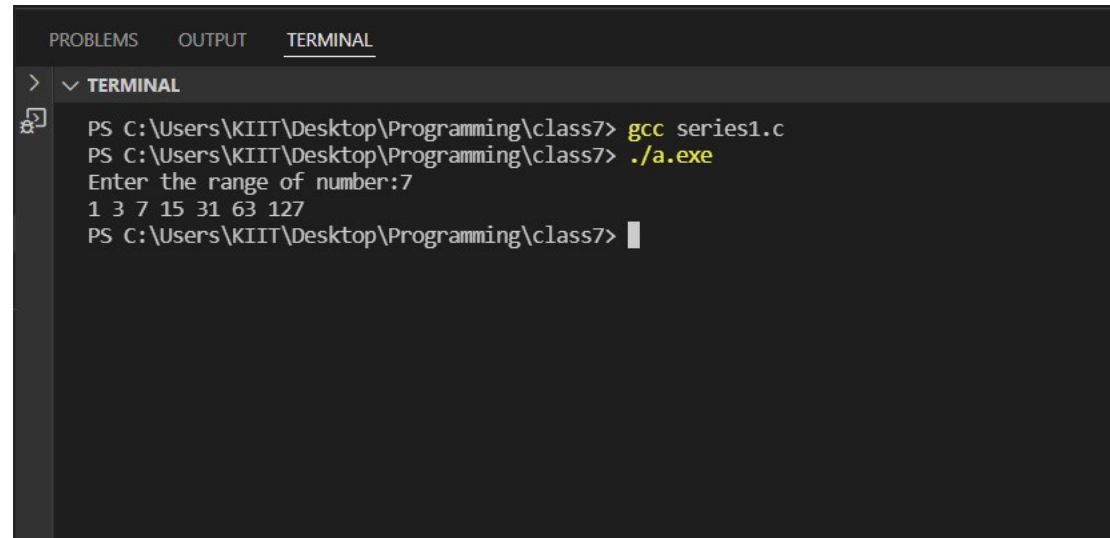
```
    {
        pr285=(pr285*2)+1;
        printf("%d ",pr285);
    }
}
```

Output:

```
PROBLEMS     OUTPUT     TERMINAL

> v TERMINAL

    PS C:\Users\KIIT\Desktop\Programming\class7> gcc series1.c
    PS C:\Users\KIIT\Desktop\Programming\class7> ./a.exe
    Enter the range of number:7
    1 3 7 15 31 63 127
    PS C:\Users\KIIT\Desktop\Programming\class7>
```

```
PS C:\Users\KIIT\Desktop\Programming\class7> ./a.exe
Enter the range of number:10
1 3 7 15 31 63 127 255 511 1023
PS C:\Users\KIIT\Desktop\Programming\class7>
```

#5. WAP to print the series as 1 1 2 3 5 8 13 ....n, where n is given by user. (Fibonacci Numbers)
Code:

```c
#include <stdio.h>
int main()
{
    int n285,a285,b285,i285;
    printf("Provide the number of fibonacci terms you want to see=>");
    a285=0; b285=1;
    int c285=a285+b285;
    scanf("%d",&n285);
    printf("%d %d %d ",a285,b285,c285);
    for(i285=3;i285<=n285;i285++)
    {
        a285=b285;
        b285=c285;
        c285=a285+b285;
        printf("%d ",c285);
    }
    return 0;
```

}

Output:

```
PROBLEMS    OUTPUT    TERMINAL

> ∨ TERMINAL

    PS C:\Users\KIIT\Desktop\Programming\class7> gcc fibonacci.c
    PS C:\Users\KIIT\Desktop\Programming\class7> ./a.exe
    Provide the number of fibonacci terms you want to see=>10
    0 1 1 2 3 5 8 13 21 34 55
    PS C:\Users\KIIT\Desktop\Programming\class7>
```

```
    PS C:\Users\KIIT\Desktop\Programming\class7> ./a.exe
    Provide the number of fibonacci terms you want to see=>5
    0 1 1 2 3 5
    PS C:\Users\KIIT\Desktop\Programming\class7>
```

#6. WAP to print the series as 3 5 7 11 13 17.........n, where n is given by user. (Prime number series)

Code;

```c
#include <stdio.h>
int main()
{
    int n285,count285,i285;
    printf("How many terms do you want to be printed?\n");
    scanf("%d",&n285);
    for(i285=1;i285<=n285;i285++)
    {
        count285=(2*i285)+1;
        printf("%d ",count285);
    }
    return 0;
}
```

Output:

```
PROBLEMS    OUTPUT    TERMINAL

∨ TERMINAL

  PS C:\Users\KIIT\Desktop\Programming\class7> gcc prime_series.c
  PS C:\Users\KIIT\Desktop\Programming\class7> ./a.exe
  How many terms do you want to be printed?
  10
  3 5 7 9 11 13 15 17 19 21
  PS C:\Users\KIIT\Desktop\Programming\class7>
```
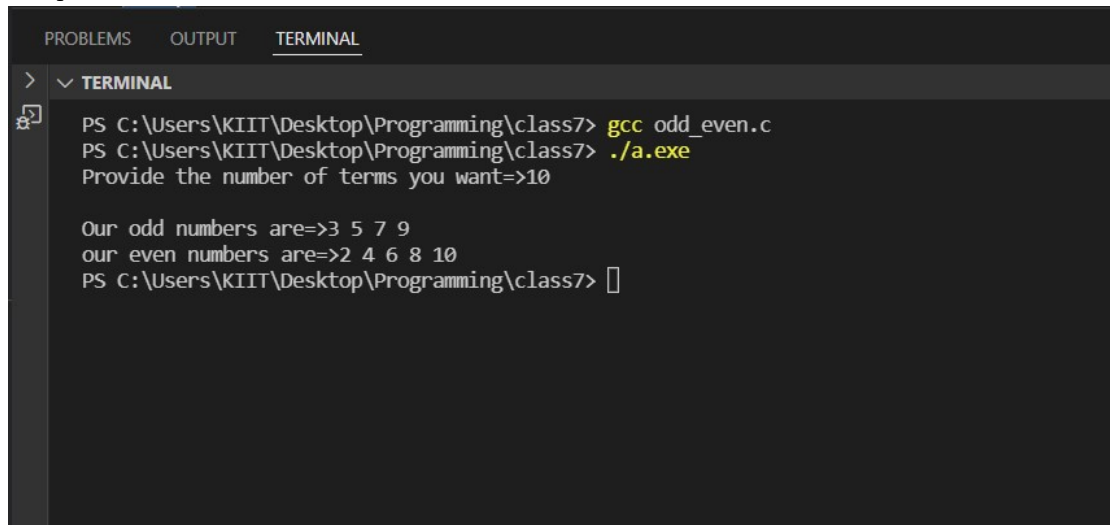
```
  PS C:\Users\KIIT\Desktop\Programming\class7> ./a.exe
  How many terms do you want to be printed?
  5
  3 5 7 9 11
  PS C:\Users\KIIT\Desktop\Programming\class7>
```

#7. WAP to print all odd and even numbers separately within a given range. The range is input through user.
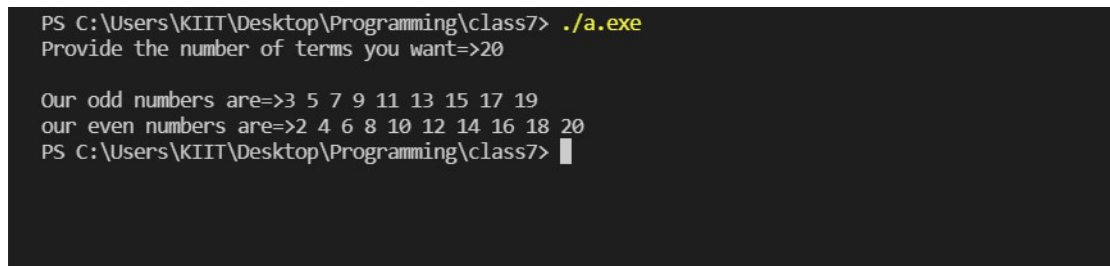Code:

```c
#include <stdio.h>
int main()
{
    int n285,count2285,count285,j285;
    printf("Provide the number of terms you want=>");
    scanf("%d",&n285);
    printf("\nOur odd numbers are=>");
    for(j285=1;j285<=n285;j285++)
    {
        count285=2*j285+1;
        if(count285<=n285)
        {
            printf("%d ",count285);
        }
    }
    printf("\nour even numbers are=>");
    for(j285=1;j285<=n285;j285++)
    {
        count2285=j285*2;
        if(count2285<=n285)
        {
            printf("%d ",count2285);
        }
    }
    return 0;
}
```

Output:

```
PROBLEMS    OUTPUT    TERMINAL

> ∨ TERMINAL

  PS C:\Users\KIIT\Desktop\Programming\class7> gcc odd_even.c
  PS C:\Users\KIIT\Desktop\Programming\class7> ./a.exe
  Provide the number of terms you want=>10

  Our odd numbers are=>3 5 7 9
  our even numbers are=>2 4 6 8 10
  PS C:\Users\KIIT\Desktop\Programming\class7> []
```
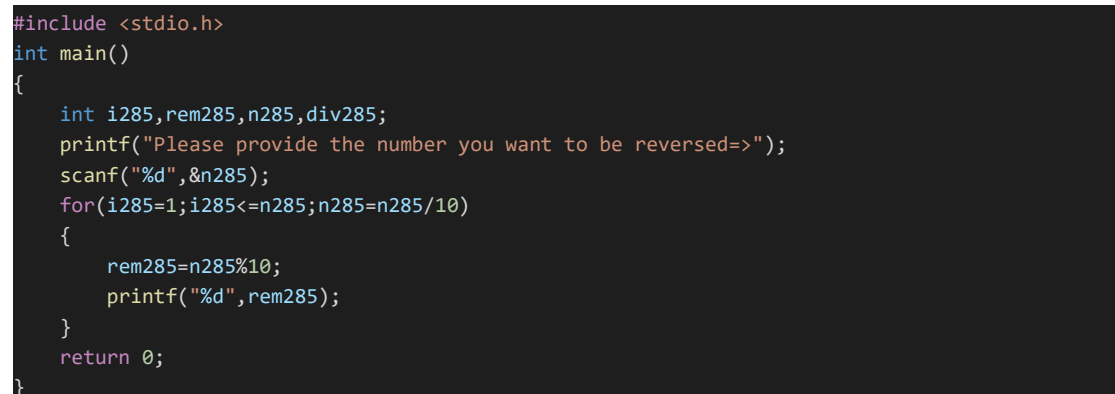
```
PS C:\Users\KIIT\Desktop\Programming\class7> ./a.exe
Provide the number of terms you want=>20

Our odd numbers are=>3 5 7 9 11 13 15 17 19
our even numbers are=>2 4 6 8 10 12 14 16 18 20
PS C:\Users\KIIT\Desktop\Programming\class7> █
```

#8. WAP to display the reverse of a number entered through keyboard.
Code;

```c
#include <stdio.h>
int main()
{
    int i285,rem285,n285,div285;
    printf("Please provide the number you want to be reversed=>");
    scanf("%d",&n285);
    for(i285=1;i285<=n285;n285=n285/10)
    {
        rem285=n285%10;
        printf("%d",rem285);
    }
    return 0;
}
```

Output:

PROBLEMS   OUTPUT   TERMINAL

∨ TERMINAL

```
PS C:\Users\KIIT\Desktop\Programming\28home> gcc reverse.c
PS C:\Users\KIIT\Desktop\Programming\28home> ./a.exe
Please provide the number you want to be reversed=>987678
876789
PS C:\Users\KIIT\Desktop\Programming\28home> ▌
```
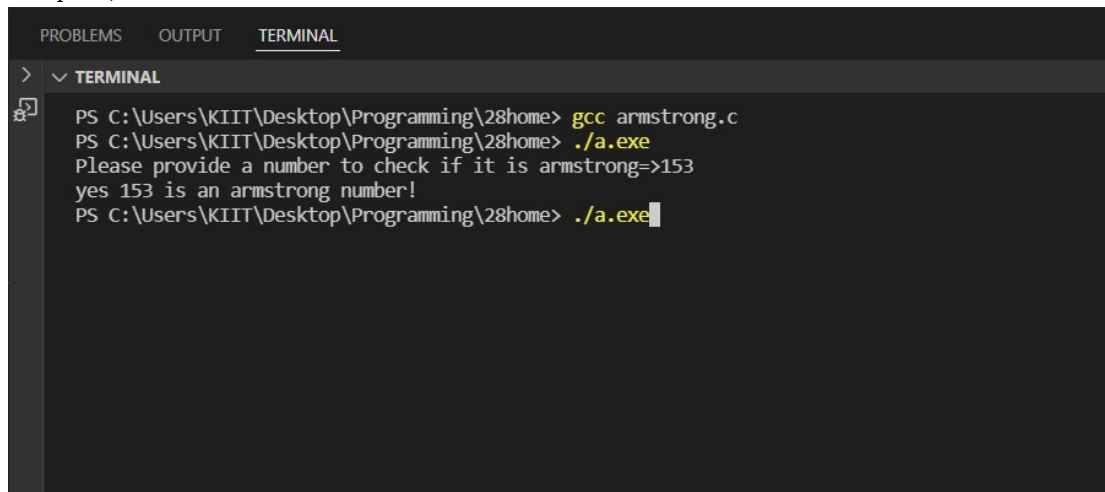
```
PS C:\Users\KIIT\Desktop\Programming\28home> ./a.exe
Please provide the number you want to be reversed=>123345
543321
PS C:\Users\KIIT\Desktop\Programming\28home> ▌
```

#9. WAP to check whether an integer number is a Armstrong number or not. (Armstrong number is a number that is equal to the sum of cubes of its digits. For example 0, 1, 153, 370, 371 and 407 are the Armstrong numbers.)
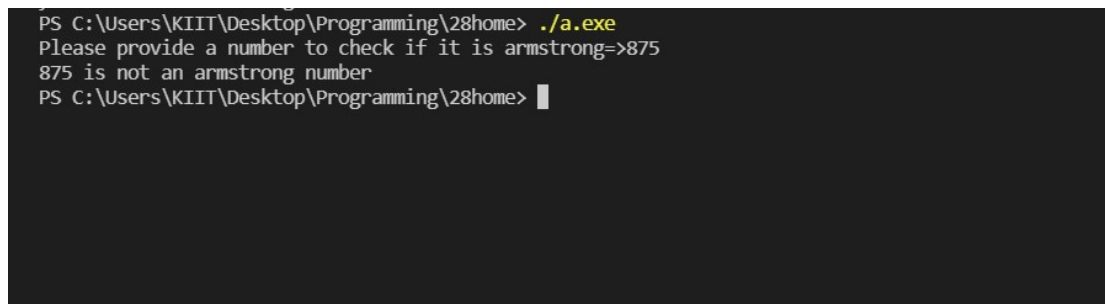
Code:

```c
#include <stdio.h>
#include <math.h>
int main()
{
    int i285,sum285,n285,rem285,globe285,j285,dummy285;
    printf("Please provide a number to check if it is armstrong=>");
    scanf("%d",&n285);
    globe285=n285;
    for(i285=1;i285<=n285;n285=n285/10)
    {
        rem285=n285%10;
        sum285= sum285 + pow(rem285,3);
        if(sum285==globe285)
        {
            dummy285=1;
            printf("yes %d is an armstrong number!\n",globe285);
        }
    }
    if(dummy285!=1)
    {
        printf("%d is not an armstrong number",globe285);
    }
    return 0;
}
```
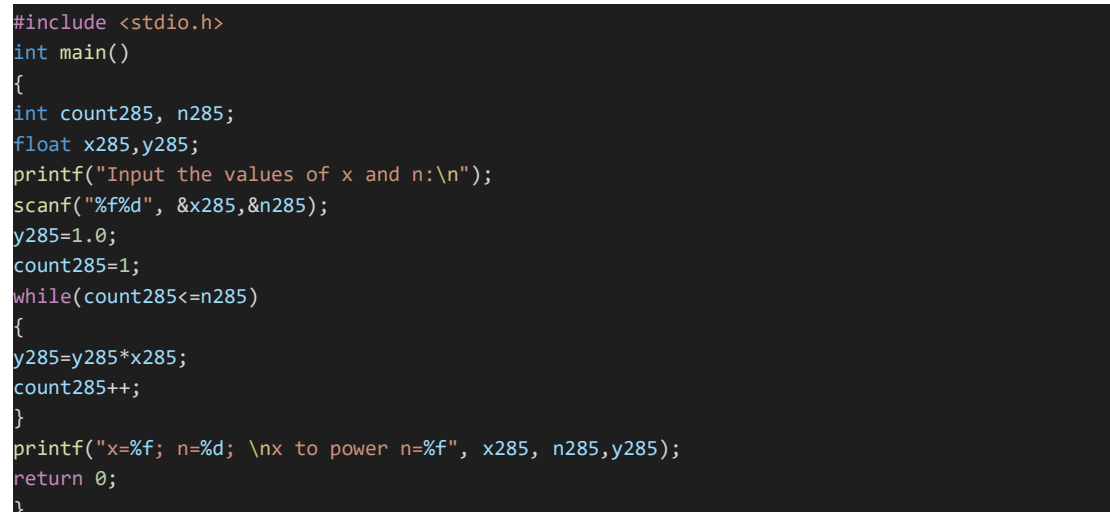
Output;

```
PROBLEMS    OUTPUT    TERMINAL

> ∨ TERMINAL
    PS C:\Users\KIIT\Desktop\Programming\28home> gcc armstrong.c
    PS C:\Users\KIIT\Desktop\Programming\28home> ./a.exe
    Please provide a number to check if it is armstrong=>153
    yes 153 is an armstrong number!
    PS C:\Users\KIIT\Desktop\Programming\28home> ./a.exe
```

```
PS C:\Users\KIIT\Desktop\Programming\28home> ./a.exe
Please provide a number to check if it is armstrong=>875
875 is not an armstrong number
PS C:\Users\KIIT\Desktop\Programming\28home>
```

#10. WAP to evaluate the equation y = x n where n is a non-negative integer

Code:

```c
#include <stdio.h>
int main()
{
int count285, n285;
float x285,y285;
printf("Input the values of x and n:\n");
scanf("%f%d", &x285,&n285);
y285=1.0;
count285=1;
while(count285<=n285)
{
y285=y285*x285;
count285++;
}
printf("x=%f; n=%d; \nx to power n=%f", x285, n285,y285);
return 0;
}
```

Output:

KIDUS ABEBE MEKONEN
Roll No- 2106285
Section A11
IT department

```
PROBLEMS    OUTPUT    TERMINAL

> ∨ TERMINAL
    PS C:\Users\KIIT\Desktop\Programming\28home> gcc square.c
    PS C:\Users\KIIT\Desktop\Programming\28home> ./a.exe
    Input the values of x and n:
    5 3
    x=5.000000; n=3;
    x to power n=125.000000
    PS C:\Users\KIIT\Desktop\Programming\28home>
```

```
    PS C:\Users\KIIT\Desktop\Programming\28home> ./a.exe
    Input the values of x and n:
    34 2
    x=34.000000; n=2;
    x to power n=1156.000000
    PS C:\Users\KIIT\Desktop\Programming\28home>
```