



VIT[®]

Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)



EMBEDDED PROGRAMMING ECE4025 (L41+L42)

Allen Ben Philipose – 18BIS0043

TASK – A

a)

```
# include<stdio.h>

int main()
{
    int i = 1;
    while (i<=10);
    {
        printf ("%d\n", i);
        i++;
    }
    return 0;
}
```

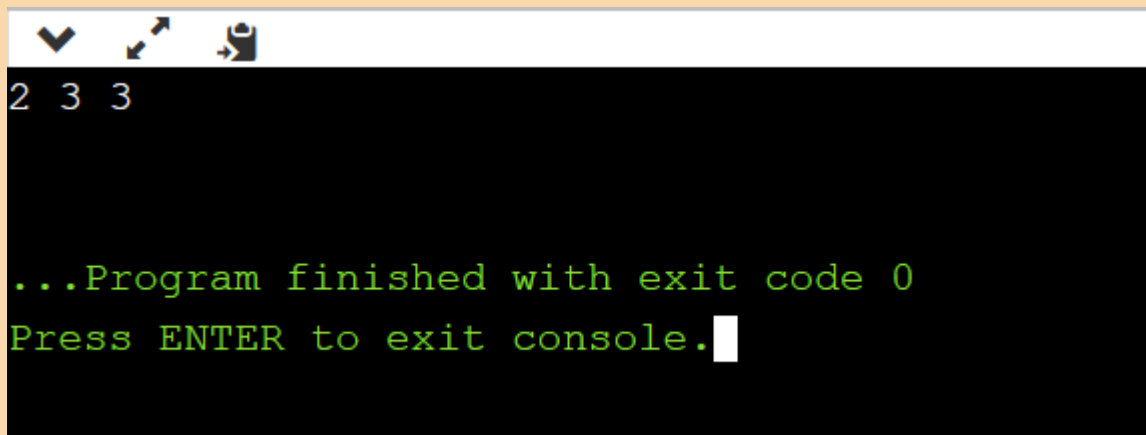


The above program will give no output, as the while statement has a semicolon (;) and the block will be ignored.

b)

```
# include<stdio.h>

int main()
{
    int x = 4, y, z;
    y = --x;
    z = x--;
    printf ("%d %d %d\n", x, y, z);
    return 0;
}
```



```
2 3 3

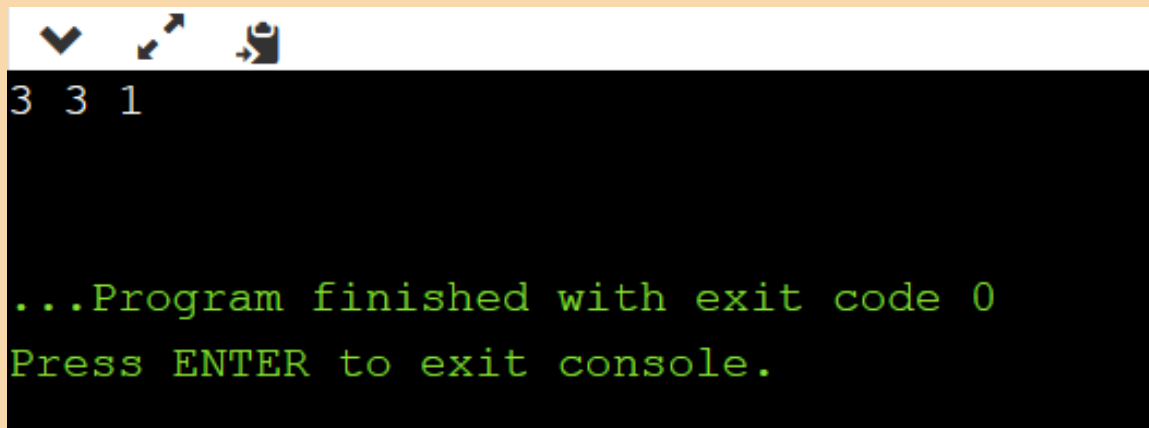
...Program finished with exit code 0
Press ENTER to exit console.
```

In this program, 'y' gets the value of 'x' after decrementing (pre-decrement) and 'z' gets the value of 'x' before decrementing the second time (post-decrement), and 'x' is decremented twice from 4.

c)

```
#include<stdio.h>

int main()
{
    int x = 4, y = 3, z;
    z = x-- - y;
    printf ("%d %d %d\n", x, y, z);
    return 0;
}
```



```
3 3 1

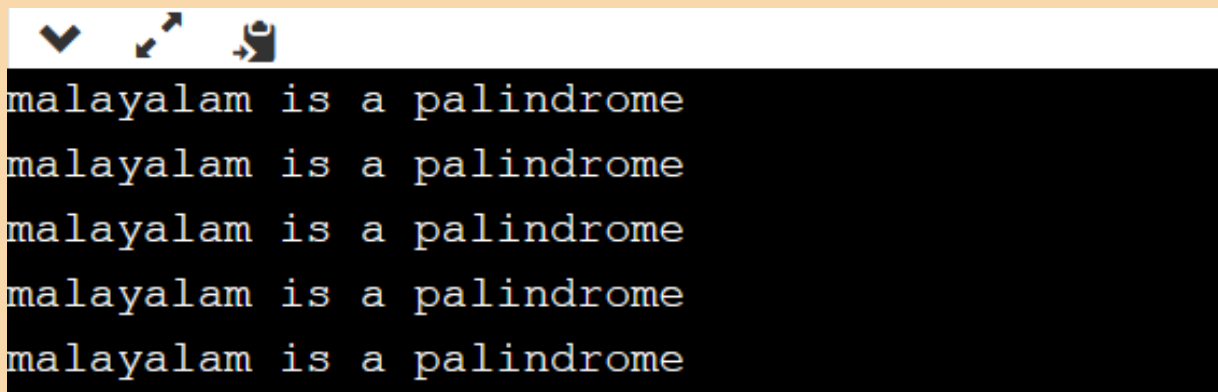
...Program finished with exit code 0
Press ENTER to exit console.
```

In this program, 'z' gets the value of 'x' before decrementing which makes the equation $4-3=1$ as the final value where 'y' is 3

d)

```
#include<stdio.h>

int main()
{
    while ('a'<'b')
        printf ("malayalam is a palindrome\n");
    return 0;
}
```

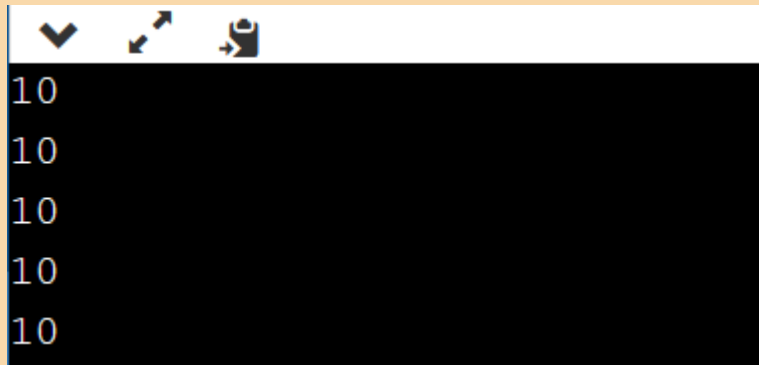


Since the character 'a' is greater than the character 'b', the loop will be infinite and the block with the print statement "Malayalam is a palindrome" will be executed infinitely.

e)

```
#include<stdio.h>

int main()
{
    int i;
    while (i=10)
    {
        printf ("%d\n", i);
        i = i+1;
    }
    return 0;
}
```

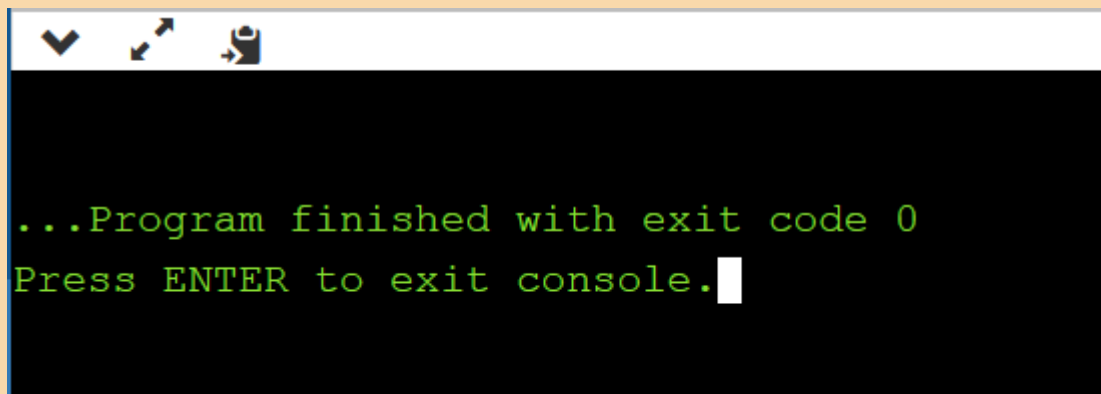


The while loop has the assignment operator (=) instead of the conditional operator (==). This makes the value of 'i' change to 10 in each iteration and hence prints the value 10 infinitely.

f)

```
#include<stdio.h>

int main()
{
    float x = 1.1;
    while (x == 1.1)
    {
        printf ("%f\n", x);
        x = x - 0.1;
    }
    return 0;
}
```

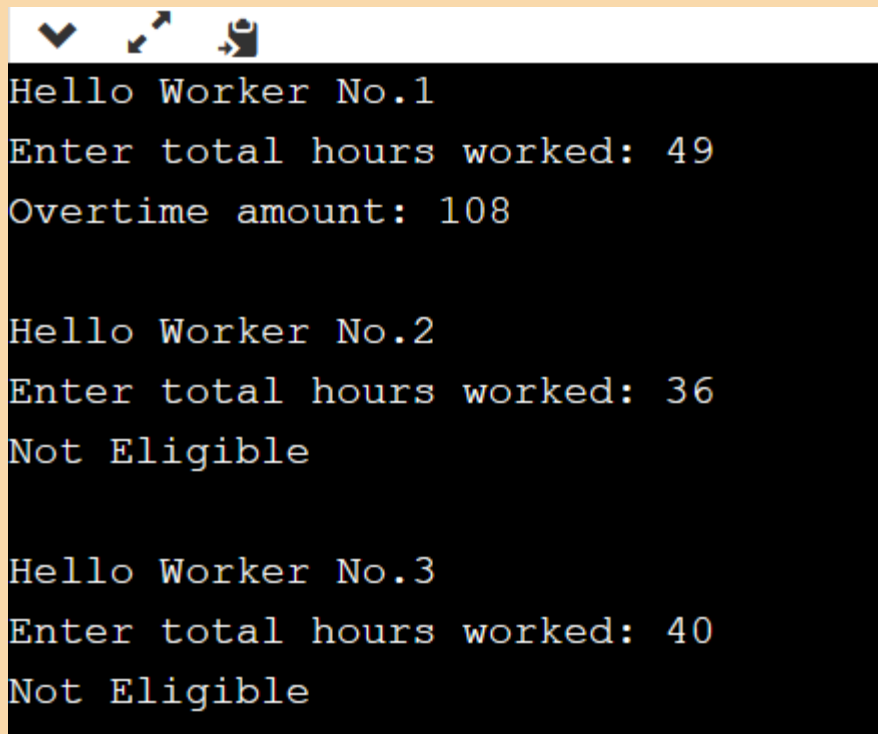


Float and Double are two different data structures and hence will be treated differently.

TASK – B

- a) **Write a program to calculate overtime pay of 10 employees. Overtime is paid at the rate of Rs. 12.00 per hour for every hour worked above 40 hours. Assume that employees do not work for fractional part of an hour.**

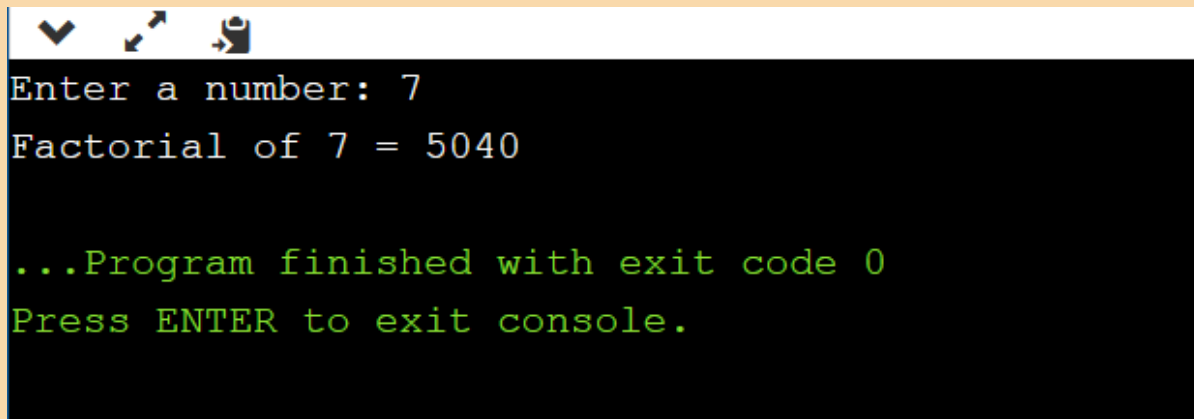
```
#include<stdio.h>
int main()
{
    for(int i=1;i<=10;i++) {
        int h;
        printf("Hello Worker No.%d\n",i);
        printf("Enter total hours worked: ");
        scanf("%d",&h);
        if(h>40) {
            printf("Overtime amount: %d\n\n",((h-
40)*12));
        }
        else {
            printf("Not Eligible\n\n");
        }
    }
    return 0;
}
```

```
✓ ↗ 📄  
Hello Worker No.1  
Enter total hours worked: 49  
Overtime amount: 108  
  
Hello Worker No.2  
Enter total hours worked: 36  
Not Eligible  
  
Hello Worker No.3  
Enter total hours worked: 40  
Not Eligible
```

b) Write a program to find the factorial value of any number entered through the keyboard.

```
#include<stdio.h>  
long int f(int n) {  
    if (n>=1)  
        return n*f(n-1);  
    else  
        return 1;  
}  
int main() {  
    int n;  
    printf("Enter a number: ");  
    scanf("%d",&n);  
    printf("Factorial of %d = %ld", n, f(n));  
    return 0;  
}
```

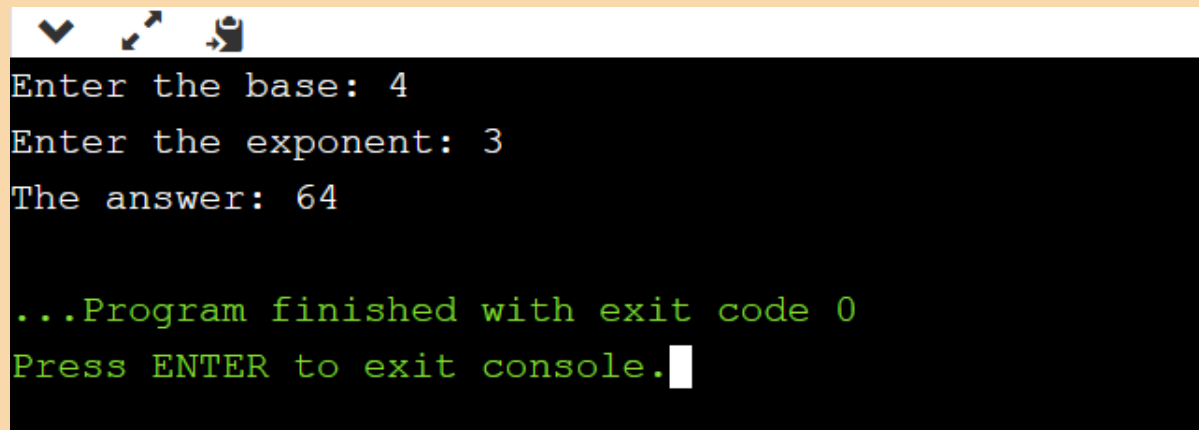


```
Enter a number: 7
Factorial of 7 = 5040

...Program finished with exit code 0
Press ENTER to exit console.
```

- c) **Two numbers are entered through the keyboard. Write a program to find the value of one number raised to the power of another.**

```
#include<stdio.h>
int main()
{
    int b,e;
    printf("Enter the base: ");
    scanf("%d",&b);
    int prod=1;
    printf("Enter the exponent: ");
    scanf("%d",&e);
    for(int i=1;i<=e;i++)
    {
        prod=prod*b;
    }
    printf("The answer: %d",prod);
    return 0;
}
```



```

Enter the base: 4
Enter the exponent: 3
The answer: 64

...Program finished with exit code 0
Press ENTER to exit console.


```

- d) Write a program to print all the ASCII values and their equivalent characters using a while loop. The ASCII values vary from 0 to 255.

```

#include<stdio.h>
int main()
{
    int i=0;
    char c;
    while(i<256) {
        printf("%c",i);
        i++;
    }
}

```



```

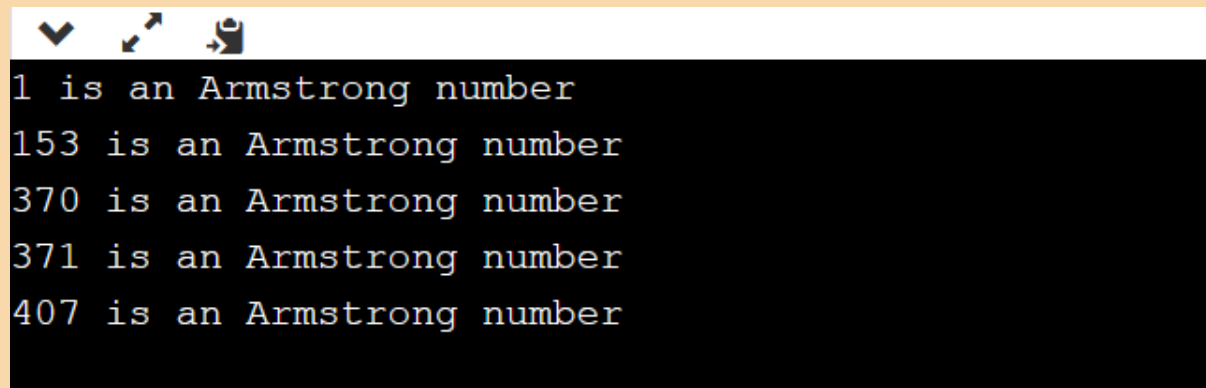
! " # $ % & ' ( ) * + , - . / 0 1 2 3 4 5 6 7 8 9 : ; < = > ? @ A B C D E F G H I J K L M N O P Q R S T U V W X Y Z [ \ ] ^ _ ` a b c d e f g h i j k l m n o p q r s t u v w x y z { | } ~ ^
...Program finished with exit code 0
Press ENTER to exit console.

```

- e) Write a program to print out all Armstrong numbers between 1 and 500. If sum of cubes of each digit of the number is equal to the number itself, then the number is called an Armstrong number. For example, $153 = (1 * 1 * 1) + (5 * 5 * 5) + (3 * 3 * 3)$.**

```
#include <stdio.h>

int main() {
    int o, r, result;
    for(int i=1;i<=500;i++) {
        o = i;
        result=0;
        while (o!=0) {
            r = o%10;
            result += r*r*r;
            o /= 10;
        }
        if (result == i)
            printf("%d is an Armstrong number \n",
i);
    }
    return 0;
}
```



```
✓ ↗ 📄  
1 is an Armstrong number  
153 is an Armstrong number  
370 is an Armstrong number  
371 is an Armstrong number  
407 is an Armstrong number
```

f) Write a program for a matchstick game being played between the computer and a user. Your program should ensure that the computer always wins. Rules for the game are as follows:


- **There are 21 matchsticks.**
- **The computer asks the player to pick 1, 2, 3, or 4 matchsticks.**
- **After the person picks, the computer does its picking.**
- **Whoever is forced to pick up the last matchstick loses the game.**

```
#include<stdio.h>  
  
int main()  
{  
    int m = 21, p, c;  
    while(1)  
    {
```

```
        printf("\nNumber of Match sticks left =
%d\n", m);
        printf("Pick 1 or 2 or 3 or 4 matches\n");
        scanf("%d", &p);
        if(p > 4 || p < 1)
            continue;

        m = m - p;
        printf("Number of matches left = %d\n",
m);

        c = 5 - p;
        printf("out of which computer picked up
%d\n", c);
        m = m - c;
        if(m == 1)
        {
            printf("\nNumber of matches left =
%d\n", m);
            printf("You lost the Game\n");
            break;
        }
    }
    return 0;
}
```



```
Number of Match sticks left = 21
Pick 1 or 2 or 3 or 4 matches
4
Number of matches left = 17
out of which computer picked up 1

Number of Match sticks left = 16
Pick 1 or 2 or 3 or 4 matches
3
```

```
Number of Match sticks left = 6
Pick 1 or 2 or 3 or 4 matches
4
Number of matches left = 2
out of which computer picked up 1

Number of matches left = 1
You lost the Game

...Program finished with exit code 0
Press ENTER to exit console.
```

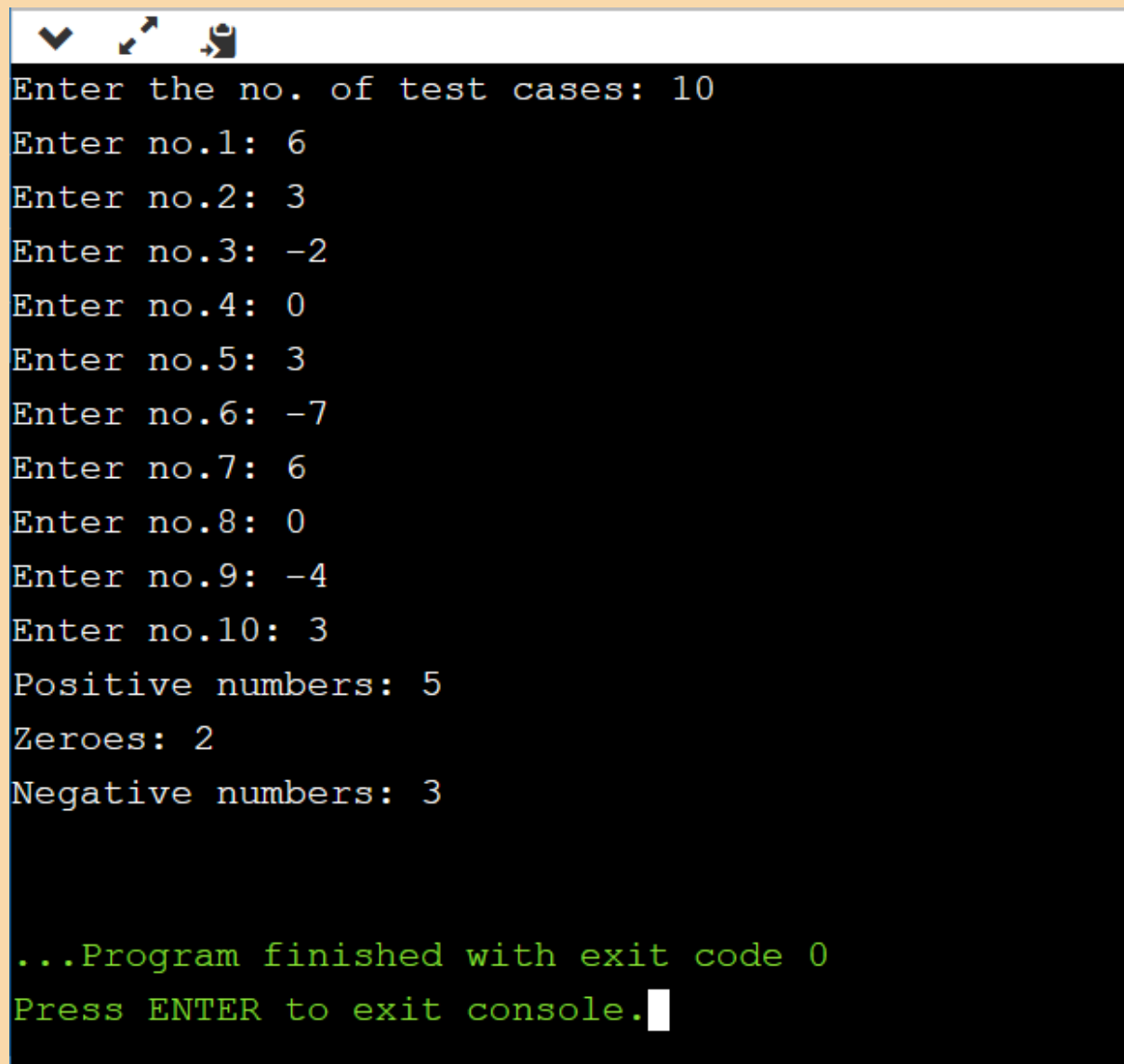
g) Write a program to enter numbers till the user wants. At the end it should display the count of positive, negative and zeros entered.

```
#include<stdio.h>

int main()
{
    int i,n,t;
    int po=0,ne=0,ze=0;
    printf("Enter the no. of test cases: ");
    scanf("%d",&n);
    for(i=0;i<n;i++)
    {
        printf("Enter no.%d: ",(i+1));
        scanf("%d",&t);
        if(t>0) {
            po++;
        }
        else if(t==0) {
            ze++;
        }
        else {
            ne++;
        }
    }
}
```



```
    }  
}  
printf("Positive numbers: %d\n",po);  
printf("Zeroes: %d\n",ze);  
printf("Negative numbers: %d\n",ne);  
return 0;  
}
```

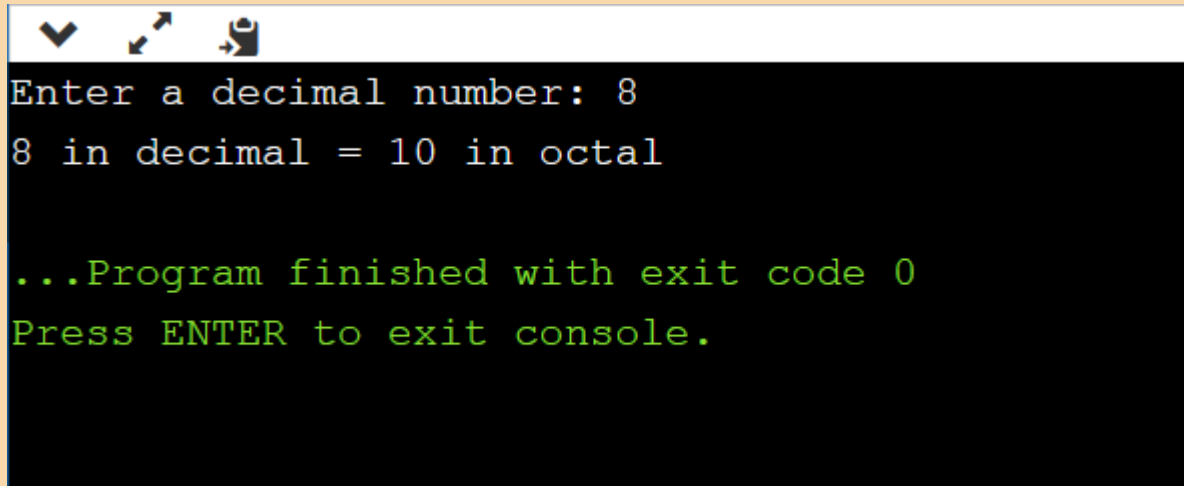
A screenshot of a terminal window with a black background and white text. The window has a title bar with standard Linux window controls (minimize, maximize, close). The text in the terminal shows the program's execution flow: it asks for the number of test cases (10), then for 10 individual numbers (6, 3, -2, 0, 3, -7, 6, 0, -4, 3). After processing, it outputs the counts: 5 positive numbers, 2 zeroes, and 3 negative numbers. The program ends with a green message indicating it finished with exit code 0 and a prompt to press ENTER to exit the console.

```
Enter the no. of test cases: 10  
Enter no.1: 6  
Enter no.2: 3  
Enter no.3: -2  
Enter no.4: 0  
Enter no.5: 3  
Enter no.6: -7  
Enter no.7: 6  
Enter no.8: 0  
Enter no.9: -4  
Enter no.10: 3  
Positive numbers: 5  
Zeroes: 2  
Negative numbers: 3  
  
...Program finished with exit code 0  
Press ENTER to exit console.
```

h) Write a program to receive an integer and find its octal equivalent.

```
#include <stdio.h>
#include <math.h>
int f(int d) {
    int o = 0, i = 1;
    while (d != 0)
    {
        o += (d % 8) * i;
        d /= 8;
        i *= 10;
    }
    return o;
}

int main() {
    int d;
    printf("Enter a decimal number: ");
    scanf("%d", &d);
    printf("%d in decimal = %d in octal", d, f(d));
    return 0;
}
```



```
Enter a decimal number: 8
8 in decimal = 10 in octal

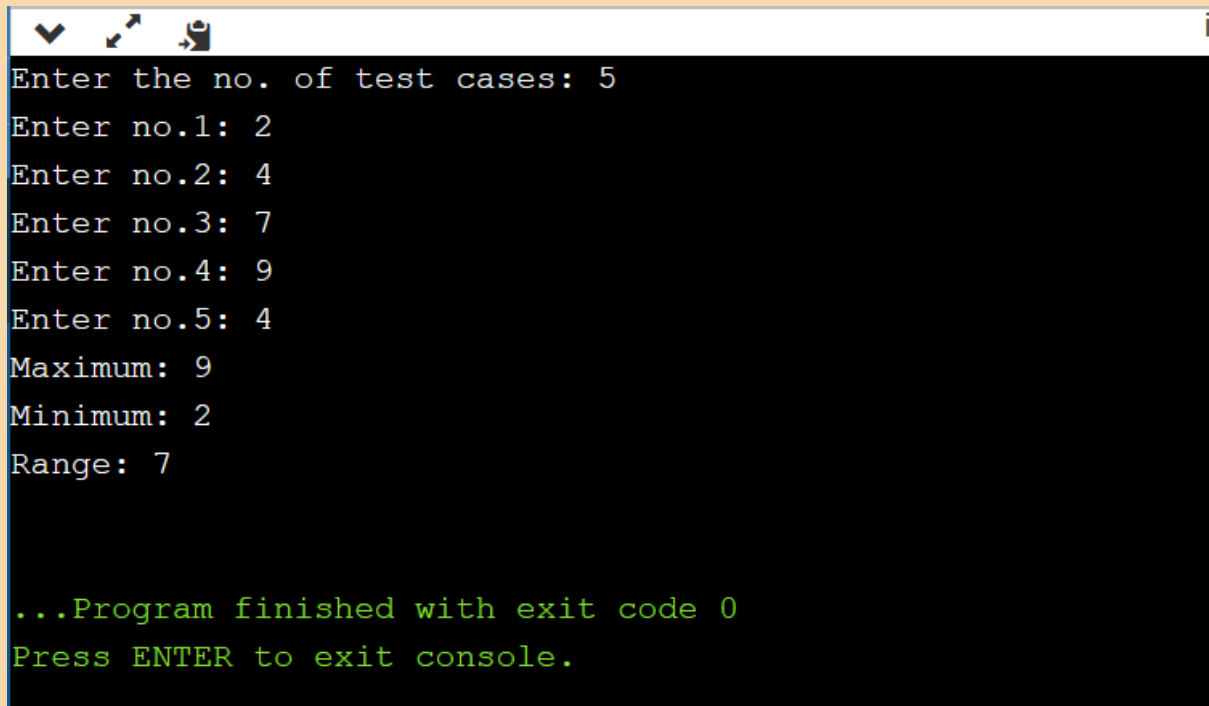
...Program finished with exit code 0
Press ENTER to exit console.
```

- i) Write a program to find the range of a set of numbers entered through the keyboard. Range is the difference between the minimum number and the maximum number.

```
#include<stdio.h>

int main()
{
    int i,n,t;
    int max=0,min=10000000;
    printf("Enter the no. of test cases: ");
    scanf("%d",&n);
    for(i=0;i<n;i++)
    {
        printf("Enter no.%d: ",(i+1));
        scanf("%d",&t);
    }
}
```

```
        if(t>max) {  
            max=t;  
        }  
        if(t<min) {  
            min=t;  
        }  
    }  
    printf("Maximum: %d\n",max);  
    printf("Minimum: %d\n",min);  
    printf("Range: %d\n",max-min);  
    return 0;  
}
```



```
Enter the no. of test cases: 5  
Enter no.1: 2  
Enter no.2: 4  
Enter no.3: 7  
Enter no.4: 9  
Enter no.5: 4  
Maximum: 9  
Minimum: 2  
Range: 7  
  
...Program finished with exit code 0  
Press ENTER to exit console.
```