

Day(02/08/2025)

1. Write a c program to add two integers.

IPO:

Input= variable x,y,z.

Process= addition (relational operator)

output= $z=x+y$

Code:

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

```
void main()
```

```
{
```

```
    Int x,y,s;
```

```
    printf("Enter first number: ");
```

```
    scanf("%d", &x);
```

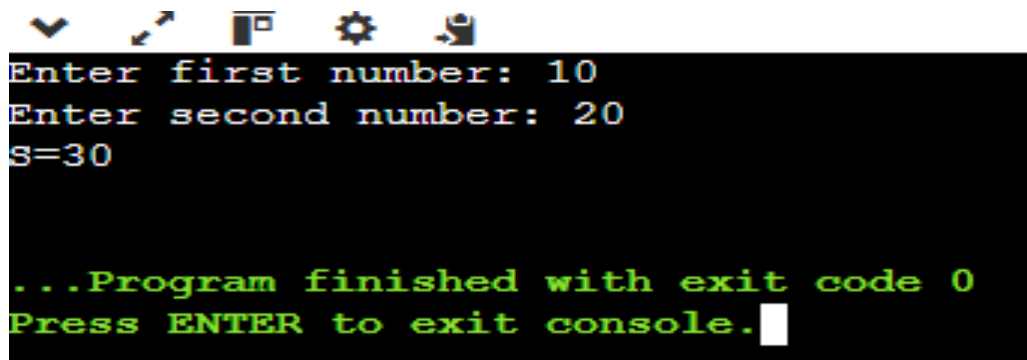
```
    printf("Enter second number: ");
```

```
    scanf("%d", &y);
```

```
    s=x+y;
```

```
    printf("S=%d\n",s);
```

```
}
```



```
Enter first number: 10
Enter second number: 20
s=30

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

2. Write a c program to swap two numbers using temporary variable.

IPO:

Input= variable a,b,c.

Process= temporary variable c is assign to a, ais assign to b, a is assign to c.

output= c=a

a=b

a=c

Code:

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

```
void main()
```

```
{
```

```
    int a,b,c;
```

```
    printf("enter the number a:");
```

```
    scanf("%d",&a);
```

```
    printf("enter the number b:");
```

```
    scanf("%d",&b);
```

```
    c=a;
```

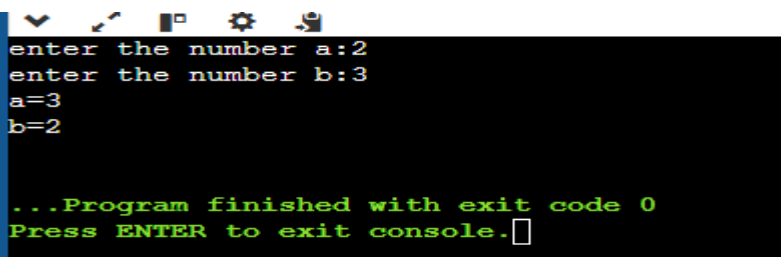
```
    a=b;
```

```
    b=c;
```

```
    printf("a=%d\n",a);
```

```
    printf("b=%d\n",b);
```

```
}
```



```
enter the number a:2
enter the number b:3
a=3
b=2
```

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

3. Write a c program to swap two numbers without temporary variable.

IPO:

Input: variable x,y

Process: a is assigned to a+b, b is assigned a-b, then a is assigned to a-b.

output: a= a+b

b=a-b

a=a-b

Code:

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

```
int main()
```

```
{
```

```
    int a,b;
```

```
    printf("Enter first number (a): ");
```

```
    scanf("%d", &a);
```

```
    printf("Enter second number (b): ");
```

```
    scanf("%d", &b);
```

```
    a = a + b;
```

```
    b = a - b;
```

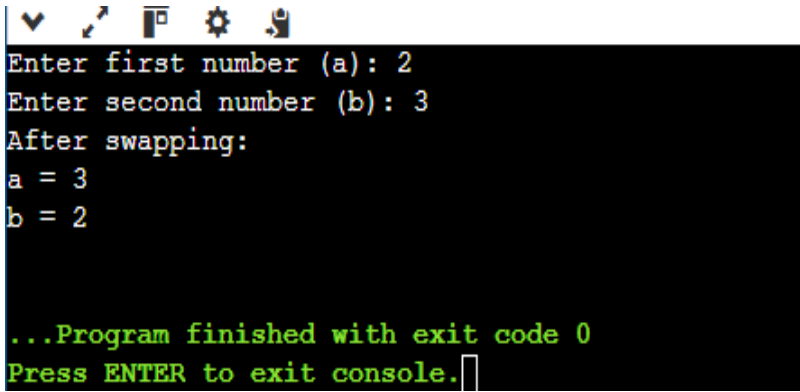
```
    a = a - b;
```

```
    printf("After swapping:\n");
```

```
    printf("a = %d\n", a);
```

```
    printf("b = %d\n", b);
```

```
}
```



```
Enter first number (a): 2
Enter second number (b): 3
After swapping:
a = 3
b = 2

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

4. Write a c program to find the ASCII value of a character.

IPO:

Input: character ch.

process: convert character into ASCII value using int char ch

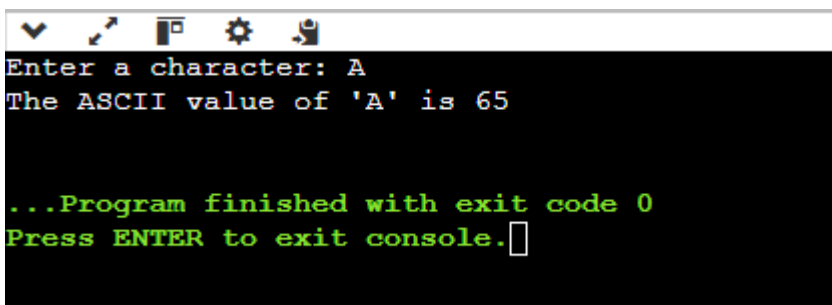
Output: display the ASCII value for a given character.

Code:

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

```
void main()
```

```
{
    char ch;
    printf("Enter a character: ");
    scanf("%c", &ch);
    printf("The ASCII value of '%c' is %d\n", ch, ch);
}
```



```
Enter a character: A
The ASCII value of 'A' is 65

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

5. Write a c program to calculate the area and perimeter of a rectangle.

IPO:

Input: variable l,w l is length ,w is width.

Process: for area is length multiply with width and for perimeter is 2 into length plus width.

Output: area(a): $l*w$.

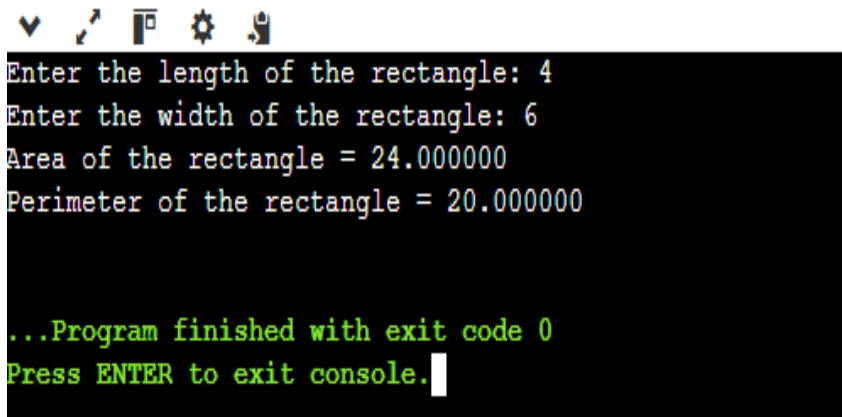
perimeter(p): $2*(l+w)$

Code:

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

```
void main()
```

```
{  
    float l, w, a, p;  
    printf("Enter the length of the rectangle: ");  
    scanf("%f", &l);  
    printf("Enter the width of the rectangle: ");  
    scanf("%f", &w);  
    a=l*w;  
    p=2*(l+w);  
    printf("Area of the rectangle = %f\n", a);  
    printf("Perimeter of the rectangle = %f\n", p);  
}
```

A terminal window with a black background and white text. At the top, there is a toolbar with icons for a dropdown menu, a cursor, a document, a gear, and a user profile. The text in the terminal shows a C program that prompts for the length and width of a rectangle, calculates the area and perimeter, and displays the results. The program has finished with exit code 0, and the user is prompted to press ENTER to exit the console.

```
Enter the length of the rectangle: 4
Enter the width of the rectangle: 6
Area of the rectangle = 24.000000
Perimeter of the rectangle = 20.000000

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

6. Write a c program to compute the simple interest.

IPO:

Input: variable p,r,t,i here p is principle, t is time, r is rate, i is interest.

Process: p is multiply t multiply r and divide by 100.

Output: $i = \frac{p \times r \times t}{100}$.

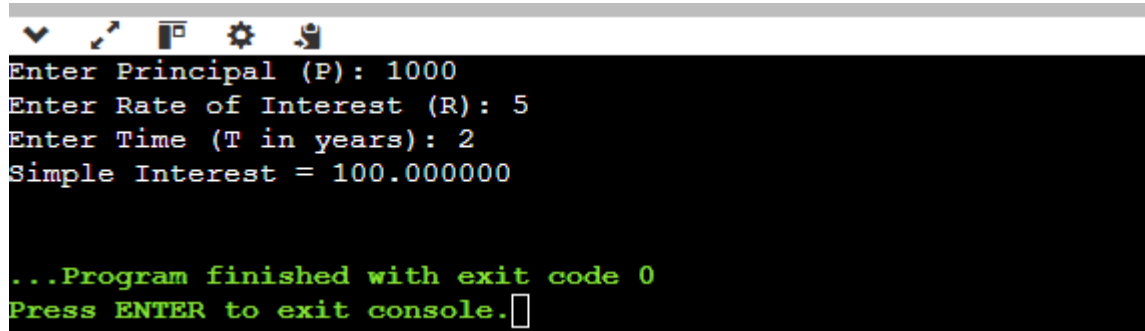
Code:

#include <stdio.h>

void main()

```
{
    float p,r,t,i;
    printf("Enter Principal (P): ");
    scanf("%f",&p);
    printf("Enter Rate of Interest (R): ");
    scanf("%f",&r);
    printf("Enter Time (T in years): ");
    scanf("%f",&t);
    i=(p*r*t)/100;
    printf("Simple Interest = %f\n", i);
}
```

}



```
Enter Principal (P): 1000
Enter Rate of Interest (R): 5
Enter Time (T in years): 2
Simple Interest = 100.000000

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

7. Write a c program to convert temperature from Celsius to Fahrenheit.

IPO:

Input: variable c,f c is celsius and f is fahrenheit.

Process= celsius is multiplied with 9/5 and added with 32.

Output: $f = (c * 9/5) + 32$.

Code:

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

```
void main()
```

```
{
```

```
    float c,f;
```

```
    printf("Enter temperature in Celsius: ");
```

```
    scanf("%f", &c);
```

```
     $f = (c * 9/5) + 32$ ;
```

```
    printf("Temperature in Fahrenheit = %.2f\n",f);
```

```
}
Enter temperature in Celsius: 37
Temperature in Fahrenheit = 98.60

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

8. Write a c program to find the quotient and remainder of two numbers.

IPO:

Input: variable a,b. q is quotient, r is remainder.

Process: for quotient a is divided by b. For remainder a modulus b.

Output: $q=a/b$, $r=a\%b$.

Code:

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

```
void main()
```

```
{
```

```
    int a,b,q,r;
```

```
    printf("Enter the number:");
```

```
    scanf("%d",&a);
```

```
    printf("Enter the number:");
```

```
    scanf("%d",&b);
```

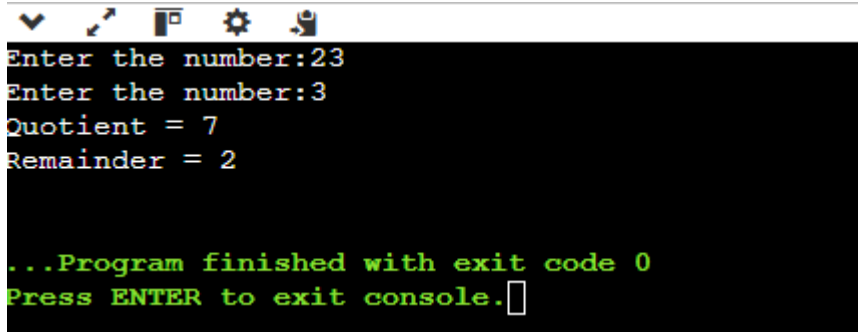
```
    q=a/b;
```

```
    r=a%b;
```

```
    printf("Quotient = %d\n",q);
```

```
    printf("Remainder = %d\n",r);
```

```
}
```

```
Enter the number:23
Enter the number:3
Quotient = 7
Remainder = 2

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

9. Write a c program to check whether a given number is odd or even.

IPO:

Input: variable n.

Process: using the if else syntax to check n modulus 2 is equal to 0 or not

Output: printf whether the given number is odd or even.

Code:

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

```
void main()
```

```
{
```

```
    int n;
```

```
    printf("Enter a number: ");
```

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

```
    if (n%2==0)
```

```
    {
```

```
        printf("%d is even.\n",n);
```

```
    }
```

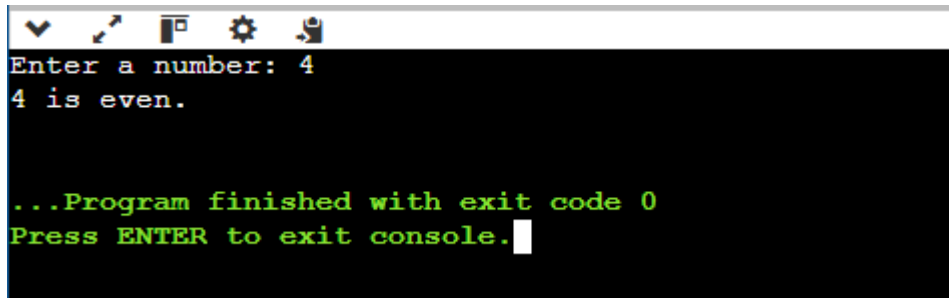
```
    else
```

```

{
    printf("%d is odd.\n",n);
}
}

```

Output for even number



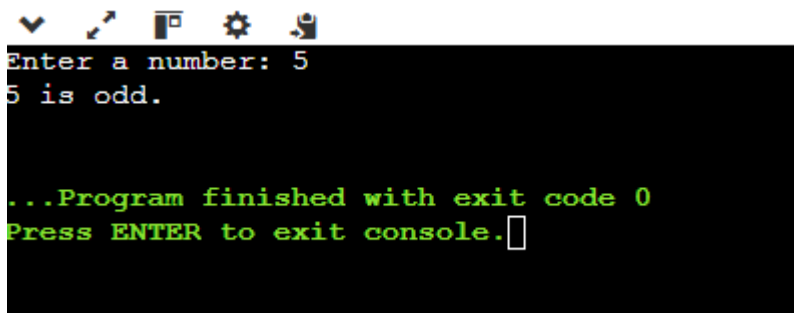
```

Enter a number: 4
4 is even.

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

```

Output for odd number



```

Enter a number: 5
5 is odd.

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

```

10. Write a c program to calculate the square and cube of a number.

IPO:

Input: variable i. Cube ,square.

Process: for square is assigned to i multiplied with i

for the cube is assigned with i multiplied i multiplied i.

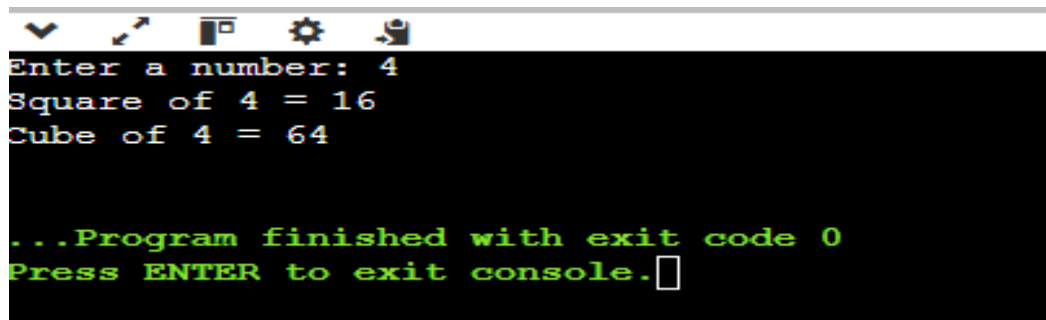
Output: square= i*i.

cube=i*i*i.

Code:

#include <stdio.h>

```
void main()
{
    int i,square, cube;
    printf("Enter a number: ");
    scanf("%d",&i);
    square=i*i;
    cube=i*i*i;
    printf("Square of %d = %d\n",i,square);
    printf("Cube of %d = %d\n",i, cube);
}
```



```
Enter a number: 4
Square of 4 = 16
Cube of 4 = 64

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