```c
/*Write a program that takes two integers as input and calculates their
sum using a function.
Pass the integers to the function using call by value.*/
#include<stdio.h>
void sum(int a,int b);
void main()
{
    int a=30,b=40;
    sum(a,b);
}
void sum(int a,int b)
{
    printf("Sum is %d",a+b);
}
```

```
PS D:\projects\quest\C> cd "d:\projects\quest\C\" ;
Sum is 70
PS D:\projects\quest\C>
```

```c
/*Write a program that takes two integers as input and calculates their sum using a function.
Pass the integers to the function using call by value.*/
#include<stdio.h>
int sum(int a,int b);
void main()
{
    int a=30,b=40,c=0;
    c=sum(a,b);
    printf("Sum is %d",c);
}
int sum(int a,int b)
{
    int s=0;
    s=a+b;
    return s;
}
```

```
PS D:\projects\quest\C> cd "d:
Sum is 70
PS D:\projects\quest\C>
```

```c
/*Implement a function that takes two integers as arguments and returns
the larger of the two.
 Demonstrate how the original values are not altered.*/
 #include<stdio.h>
 void swap(int a,int b);
 void main()
 {
    int a=10,b=5;
    printf("Value of a is %d and value of b is %d\n",a,b);
    swap(a,b);
    printf("Value of a is %d and value of b is %d\n",a,b);
 }
 void swap(int a,int b)
 {
    int t;
    t=a;
    a=b;
    b=t;
    printf("Value of a is %d and value of b is %d\n",a,b);
 }
 //original values are not altered as a copy of the original values are
fed as input to the function
```

```
PS D:\projects\quest\C> cd "d:\projects\q
Value of a is 10 and value of b is 5
Value of a is 5 and value of b is 10
Value of a is 10 and value of b is 5
PS D:\projects\quest\C>
```

```c
/*Implement a function that takes two integers as arguments and returns
the larger of the two.
```

```c
Demonstrate how the original values are not altered.*/
#include<stdio.h>
void max(int a,int b)
{
   if(a>b)
   printf("%d",a);
   else
   printf("%d",b);
}
void main()
{
   int a=30,b=45;
   max(a,b);
}
```

```
PS D:\projects\quest\C> cd "d:\project
45
PS D:\projects\quest\C>
```

```c
/*Implement a function that takes two integers as arguments and returns
the larger of the two.
 Demonstrate how the original values are not altered.*/
 #include<stdio.h>
int max(int a,int b)
{
   if(a>b)
   return a;
   else
   return b;
}
void main()
{
   int a=30,b=45,c;
   c=max(a,b);
   printf("%d",c);
}
```

```
PS D:\projects\quest\C> cd "d:\project
45
PS D:\projects\quest\C>
```

```c
/*Create a function to compute the factorial of a given number passed to
it.
Ensure the original number remains unaltered.*/
#include<stdio.h>
void fact(int n)
{
    int f=1;
    for(int i=1;i<=n;i++)
    f=f*i;
    printf("The factorial of %d is %d",n,f);
}
void main()
{
    int n=7;
    fact(7);
}
```

```
PS D:\projects\quest\C> cd "d:\p
The factorial of 7 is 5040
PS D:\projects\quest\C>
```

```c
/*Create a function to compute the factorial of a given number passed to
it.
Ensure the original number remains unaltered.*/
#include<stdio.h>
int fact(int n)
{
    int f=1;
    for(int i=1;i<=n;i++)
    f=f*i;
    return f;
}
void main()
{
    int f;
```

```
    int n=7;
    f=fact(n);
    printf("Fctorial of %d is %d",n,f);
}
```

```
PS D:\projects\quest\C> cd "d:\
Fctorial of 7 is 5040
PS D:\projects\quest\C>
```

```
/*
Write a program where a function determines whether a given integer is
even or odd. The function should use call by value.
*/
#include<stdio.h>
void check(int n)
{
    if(n%2==0)
    printf(" %d is an even number\n",n);
    else
    printf("%d is an odd number",n);
}
void main()
{
    int n=7;
    check(n);
}
```

```
PS D:\projects\quest\C> cd "d:\
7 is an odd number
PS D:\projects\quest\C>
```

```
/*
Write a program where a function determines whether a given integer is
even or odd. The function should use call by value.
*/
#include<stdio.h>
```

```c
int check(int n)
{
    if(n%2==0)
    return 0;
    else
    return 1;
}
void main()
{
    int n=7,f;
    f=check(n);
    if(f==0)
    printf("%d is even",n);
    else
    printf("%d is odd",n);
}
```

```
PS D:\projects\quest\C> cd "d:\projec
7 is odd
PS D:\projects\quest\C>
```

```c
/*Write a program that calculates simple interest using a function.
Pass principal, rate, and time as arguments and return the computed
interest.*/
#include<stdio.h>
void interest(int p,int r,int t);
void main()
{
    int p=2400,r=5,t=10;
    interest(p,r,t);
}
void interest(int p,int r,int t)
{
    int si;
```

```
    si=(p*r*t)/100;
    printf("SI is %d",si);
}
```

```
PS D:\projects\quest\C> cd "d:\p
SI is 1200
PS D:\projects\quest\C>
```

```
/*Write a program that calculates simple interest using a function.
Pass principal, rate, and time as arguments and return the computed
interest.*/
#include<stdio.h>
int interest(int p,int r,int t);
void main()
{
    int p=2400,r=5,t=10,si;
    si=interest(p,r,t);
    printf("SI is %d",si);
}
int interest(int p,int r,int t)
{
    int si;
    si=(p*r*t)/100;
    return si;
}
```

```
PS D:\projects\quest\C> cd "d:\projec
SI is 1200
PS D:\projects\quest\C>
```

```
/*Create a function that takes an integer and returns its reverse.
Demonstrate how call by value affects the original number.*/
#include<stdio.h>
void reverse(int n)
{
    int r,rev=0;
    while(n%10!=0)
```

```
    {
        r=n%10;
        n=n/10;
        rev=rev*10+r;

    }
    printf("reverse is %d",rev);
}
void main()
{
    int n=123;
    reverse(n);
}
```

```
PS D:\projects\quest\C> cd "d:\projects\q
reverse is 321
PS D:\projects\quest\C>
```

```
/*Create a function that takes an integer and returns its reverse.
Demonstrate how call by value affects the original number.*/
#include<stdio.h>
int reverse(int n)
{
    int r,rev=0;
    while(n%10!=0)
    {
        r=n%10;
        n=n/10;
        rev=rev*10+r;

    }
    return rev;
}
void main()
{
    int n=123,rev;
    rev=reverse(n);
    printf("Reverse of %d is %d",n,rev);
}
```

```
PS D:\projects\quest\C> cd "d:\proje
Reverse of 123 is 321
PS D:\projects\quest\C>
```

```c
/*Write a function to calculate the greatest common divisor (GCD) of two
numbers passed by value*/
#include<stdio.h>
void gcd(int a,int b)
{
    int gcd;
    for(int i=1;i<=a && i<=b;i++)
    {
        if(a%i==0 && b%i==0)
        gcd=i;
    }
    printf("GCD is %d",gcd);
}
void main()
{
    int a=16,b=12;
    gcd(a,b);
}
```

```
PS D:\projects\quest\C> cd "d:\
GCD is 4
PS D:\projects\quest\C>
```

```c
/*Write a function to calculate the greatest common divisor (GCD) of two
numbers passed by value*/
#include<stdio.h>
int gcd(int a,int b)
{
    int gcd;
    for(int i=1;i<=a && i<=b;i++)
```

```
    {
        if(a%i==0 && b%i==0)
        gcd=i;
    }
    return gcd;
}
void main()
{
    int a=16,b=12,c;
    c=gcd(a,b);
    printf("GCD is %d",c);
}
```

```
PS D:\projects\quest\C> cd "d:\p
GCD is 4
PS D:\projects\quest\C>
```

```
/*Implement a function that computes the sum of the digits of a number
passed as an argument.*/
#include<stdio.h>
void sum(int n)
{
    int s=0,r=0;
    while(n%10!=0)
    {
        r=n%10;
        n=n/10;
        s+=r;
    }
    printf("Sum is %d",s);
}
void main()
{
    int n=123;
    sum(123);
}
```

```
PS D:\projects\quest\C> cd "d:\projects
Sum is 6
PS D:\projects\quest\C>
```

```c
/*Implement a function that computes the sum of the digits of a number
passed as an argument.*/
#include<stdio.h>
int sum(int n)
{
    int s=0,r=0;
    while(n%10!=0)
    {
        r=n%10;
        n=n/10;
        s+=r;
    }
    return s;
}
void main()
{
    int n=123,s;
    s=sum(123);
    printf("Sum is %d",s);
}
```

```
PS D:\projects\quest\C> cd "d:\p
Sum is 6
PS D:\projects\quest\C>
```

```c
/*Write a program where a function checks if a given number is prime. Pass
the number as an argument by value.*/
#include<stdio.h>
void prime(int n)
{
    int f=1;
    if(n==2)
```

```
    printf("It is a prime number");
    else
    {
    for(int i=3;i<n;i++)
    {
        if(n%i==0)
        f=0;
    }
    }
    if(f==1)
    printf("It is a prime number");
    else
    printf("It is not a prime number");
}
void main()
{
    int n=7;
    prime(n);
}
```

```
PS D:\projects\quest\C> cd "d
It is a prime number
PS D:\projects\quest\C>
```

```
/*Write a program where a function checks if a given number is prime. Pass
the number as an argument by value.*/
#include<stdio.h>
int prime(int n)
{
    int f=1;
    if(n==2)
    return 1;
    else
    {
    for(int i=3;i<n;i++)
    {
        if(n%i==0)
        {
            f=0;
        return 0;
```

```
        }
    }
    }
    if(f==1)
    return 1;
}
void main()
{
    int n=7,f;
    f=prime(n);
    if(f==1)
    printf("It is prime");
    else
    printf("It is not a prime");
}
```

```
PS D:\projects\quest\C> cd "d:\pr
It is prime
PS D:\projects\quest\C>
```

```
/*Create a function that checks whether a given number belongs to the
Fibonacci sequence. Pass the number by value.*/
#include<stdio.h>
#include<math.h>
void fibo(int n)
{
    int sq1=0,sq2=0,q1,q2;
    q1=((5*pow(n,2))+4);
    q2=((5*pow(n,2))-4);
    sq1=sqrt(q1);
    sq2=sqrt(q2);
    if(q1%sq1==0 || q2%sq2==0)
    printf("It belongs to fibonacci");
    else
    printf("It does not belong to fibonacci");
}
void main()
{
    int n=5;
```

```
    fibo(n);
}
```

```
PS D:\projects\quest\C> cd
It belongs to fibonacci
PS D:\projects\quest\C>
```

```
/*Create a function that checks whether a given number belongs to the
Fibonacci sequence. Pass the number by value.*/
#include<stdio.h>
#include<math.h>
int fibo(int n)
{
    int sq1=0,sq2=0,q1,q2;
    q1=((5*pow(n,2))+4);
    q2=((5*pow(n,2))-4);
    sq1=sqrt(q1);
    sq2=sqrt(q2);
    if(q1%sq1==0 || q2%sq2==0)
    return 1;
    else
    return 0;
}
void main()
{
    int n=5,f;
    f=fibo(n);
      if(f==1)
    printf("It belongs to fibonacci");
    else
    printf("It does not belong to fibonacci");
}
```

```
PS D:\projects\quest\C> cd "d:\
It belongs to fibonacci
PS D:\projects\quest\C>
```

```
/*Write a function to calculate the roots of a quadratic equation
ax2+bx+c=0ax^2 + bx + c = 0 ax2+bx+c=0.
 Pass the coefficients a,b,a, b,a,b, and ccc as arguments.*/
```

```c
#include<stdio.h>
#include<math.h>
void quad(int a, int b,int c)
{
    int x1=0,x2=0,y1=0,y2=0;
    x1=(-b-(sqrt(pow(b,2)-(4*a*c))))/(2*a);
    x2=(-b+(sqrt(pow(b,2)-(4*a*c))))/(2*a);
    printf("x1 = %d,x2 =%d",x1,x2);

}
void main()
{
    int a=1,b=6,c=8;
    quad(a,b,c);
}
```

```
PS D:\projects\quest\C> cd "d:\pr
x1 = -4,x2 =-2
PS D:\projects\quest\C>
```

```c
/*Implement a function to convert a binary number (passed as an integer)
into its decimal equivalent.*/
#include<stdio.h>
#include<math.h>
void dec(int n)
{
    int r,d=0,i=0;
    while(n>0)
    {
        r=n%10;
        n=n/10;
        d+=r*pow(2,i);
        i++;
    }
    printf("Decimal is %d",d);
}
void main()
{
    int binary=111;
```

```
    dec(binary);
}
```

```
PS D:\projects\quest\C> cd "d:\projects\ques
Decimal is 7
PS D:\projects\quest\C>
```

```c
/*Implement a function to convert a binary number (passed as an integer)
into its decimal equivalent.*/
#include<stdio.h>
#include<math.h>
int dec(int n)
{
    int r,d=0,i=0;
    while(n>0)
    {
        r=n%10;
        n=n/10;
        d+=r*pow(2,i);
        i++;
    }
    return d;
}
void main()
{
    int binary=111,d;
    d=dec(binary);
    printf("Decimal is %d",d);
}
```

```
PS D:\projects\quest\C> cd "d:\
Decimal is 7
PS D:\projects\quest\C>
```

```c
/*Write a program where a function computes the trace of a 2x2 matrix (sum
of its diagonal elements).
Pass the matrix elements individually as arguments.*/
#include<stdio.h>
void sum(int a,int b)
```

```
{
    int s;
    s=a+b;
    printf("Trace is %d",s);
}
 void main()
 {
    int ar[2][2]={{1,2},
    {3,4}};
    sum(ar[0][0],ar[1][1]);
 }
```

```
PS D:\projects\quest\C> cd "d:\pr
Trace is 5
PS D:\projects\quest\C>
```

```
/*Write a program where a function computes the trace of a 2x2 matrix (sum
of its diagonal elements).
Pass the matrix elements individually as arguments.*/
#include<stdio.h>
int sum(int a,int b)
{
    int s;
    s=a+b;
    return s;
}
 void main()
 {
    int ar[2][2]={{1,2},
    {3,4}};
    int s;
   s = sum(ar[0][0],ar[1][1]);
   printf("%d",s);
 }
```

```
PS D:\projects\quest\C> cd "d:\projects\qu
5
PS D:\projects\quest\C>
```

```c
/*Create a function that checks whether a given number is a palindrome.
Pass the number by value and return the result.*/
#include<stdio.h>
int pall(int n)
{
    int rev=0,r;

    while(n!=0)
    {
        r=n%10;
        n=n/10;
        rev=rev*10+r;
    }
    return rev;
}
void main()
{
    int n=121,m;
    m=pall(n);
    if(m==n)
    printf("Pallindrome");
    else
    printf("Not a pallindrome");
}
```

```
PS D:\projects\quest\C> cd "d:\pr
Pallindrome
PS D:\projects\quest\C>
```

```c
/*Input: A floating-point value representing the measurement and a
character
indicating the conversion type (e.g., 'C' for cm-to-inches or 'I' for
inches-to-cm).
Output: The converted value.
Function:
float convert_units(float value, char type);*/
#include<stdio.h>
float convert(float value,char type)
{
    if(type =='C')
    printf("%f cm is %f inch",value,value/2.54);
    else if(type=='I')
    printf("%f inch is %f cm",value,value*2.54);
}
void main()
{
    float value=0;
    char type;
    printf("Enter value and conversion\n");
    scanf("%f %c",&value,&type);
    convert(value,type);
}
```

```
PS D:\projects\quest\C> cd "d:\projects\quest
Enter value and conversion
20 C
20.000000 cm is 7.874016 inch
PS D:\projects\quest\C>
```

```c
/*Input: Two integers: the total length of the raw material and the
desired length of each piece.
Output: The maximum number of pieces that can be cut and the leftover
material.
Function:
int calculate_cuts(int material_length, int piece_length);*/
#include<stdio.h>
int cuts(int l,int dl)
{
```

```
    printf("maximum number of pieces is %d\n",l/dl);
    printf("Leftover material is %d",l%dl);
}
void main()
{
    int l,dl;
    printf("Enter the total length and desired length\n");
    scanf("%d %d",&l,&dl);
    cuts(l,dl);
}
```

```
PS D:\projects\quest\C> cd "d:\projects\ques
Enter the total length and desired length
20 4
maximum number of pieces is 5
Leftover material is 0
PS D:\projects\quest\C> cd "d:\projects\ques
Enter the total length and desired length
20 3
maximum number of pieces is 6
Leftover material is 2
PS D:\projects\quest\C>
```

```
/*Input: Two floating-point numbers: belt speed (m/s) and pulley diameter
(m).
Output: The RPM of the machine.
Function:
float calculate_rpm(float belt_speed, float pulley_diameter);*/
#include<stdio.h>
#define pi 3.14
float calculate_rpm(float s,float d)
{
    float rpm;
    rpm=(s*60)/(pi*d);
    printf("RPM is %f",rpm);
}
void main(){
    float belt_speed,pulley_diameter;
```

```
    printf("Enter the belt speed and pulley diameter\n");
    scanf("%f %f",&belt_speed,&pulley_diameter);
    calculate_rpm(belt_speed,pulley_diameter);
}
```

```
PS D:\projects\quest\C> cd "d:\projects\quest
Enter the belt speed and pulley diameter
20 10
RPM is 38.216560
PS D:\projects\quest\C>
```

```
/*Input: Two integers: machine speed (units per hour) and efficiency
(percentage).
Output: The effective production rate.
Function:
int calculate_production_rate(int speed, int efficiency);*/
#include<stdio.h>
int rate(int s,int e)
{
    int p;
    p=(s*e)/100;
    printf("production rate is %d",p);
}
void main()
{
    int e,s;
    printf("Enter speed and efficiency\n");
    scanf("%d %d",&s,&e);
    rate(s,e);
}
```

```
PS D:\projects\quest\C> cd "d:\projects\qu
Enter speed and efficiency
100 30
production rate is 30
PS D:\projects\quest\C>
```

```c
/*Input: Two integers: total material length and leftover material length.
Output: The amount of material wasted.
Function:
int calculate_wastage(int total_length, int leftover_length);*/
#include<stdio.h>
int wastage(int tl,int ll)
{
    int used;
    used=tl-ll;

    printf("Wastage is %d",used%5);
    return 0;

}
void main()
{
    int length,leftover;
    printf("Enter length and leftover length\n");
    scanf("%d %d",&length,&leftover);
    wastage(length,leftover);
}
```

```
PS D:\projects\quest\C> cd "d:\proj
Enter length and leftover length
30 4
Wastage is 1
PS D:\projects\quest\C>
```

```c
/*Input: Three floating-point numbers: power rating (kW), operating hours,
and cost per kWh.
Output: The total energy cost.
Function:
float calculate_energy_cost(float power_rating, float hours, float
cost_per_kwh);*/
#include<stdio.h>
void cost(float pr,float h,float c)
{
    printf("Total energy cost is %f",pr*h*c);
}
void main()
{
    float power_rating,hours,cost_per_kwh;
    printf("Enter the power rating hours used and rate\n");
    scanf("%f %f %f",&power_rating,&hours,&cost_per_kwh);
    cost(power_rating,hours,cost_per_kwh);
}
```

```
PS D:\projects\quest\C> cd "d:\projects\quest\C
Enter the power rating hours used and rate
200 30 6
Total energy cost is 36000.000000
PS D:\projects\quest\C>
```

```c
/*Input: Two floating-point numbers: power usage (Watts) and efficiency
(%).
Output: Heat generated (Joules).
Function:
float calculate_heat(float power_usage, float efficiency);*/
#include<stdio.h>
float heat(float u,float e)
{
    float heat,t=10;
    printf("Here time is 10\n");
    heat=(u*(1-e))*t;
    printf("Heat generated is %f",heat);
}
void main()
{
```

```c
    float usage,efficiency;
    printf("Enter the power usage and efficiency\n");
    scanf("%f %f",&usage,&efficiency);
    heat(usage,efficiency);
}
```

```
PS D:\projects\quest\C> cd "d:\projec
Enter the power usage and efficiency
200 .7
Here time is 10
Heat generated is 600.000000
PS D:\projects\quest\C>
```

```c
/*Input:Input: A floating-point number for operating time (hours) and an
integer for material type (e.g., 1 for metal, 2 for plastic).
Output: Wear rate (percentage).
Function:
float calculate_wear_rate(float time, int material_type);
Function*/
#include<stdio.h>
float wear(float time,int type)
{
    float wp=.4,wm=.7;
    if(type==1)
    {
        printf("Material is metal,wear rate is %f\n",((1-wm)/1)*100);

    }
    else if(type == 2)
    {
        printf("Material is plastic,wear rate is %f\n",((1-wp)/1)*100);
    }
}
void main()
{
    float time;
    int type;
```

```
    printf("Enter the time and material type(1 for metal 2 for
plastic)\n");
    scanf("%f %d",&time,&type);
    wear(time,type);
}
```

```
PS D:\projects\quest\C> cd "d:\projects\quest\C\
Enter the time and material type(1 for metal 2 f
20 1
Material is metal,wear rate is 30.000001
PS D:\projects\quest\C>
```

```
/*Input: Two integers: consumption rate (units/day) and lead time (days).
Output: Reorder quantity (units).
Function:
int calculate_reorder_quantity(int consumption_rate, int lead_time);*/
#include<stdio.h>
int reorder(int cr,int ld)
{
    printf("Reorder quantity is %d\n",cr*ld);
}
void main()
{
    int cr,lt;
    printf("Enter the consumption rate and lead time\n");
    scanf("%d %d",&cr,&lt);
    reorder(cr,lt);

}
```

```
PS D:\projects\quest\C> cd "d:\projec
Enter the consumption rate and lead t
20 50
Reorder quantity is 1000
PS D:\projects\quest\C>
```

```
/*Input: Two integers: number of defective items and total batch size.
Output: Defective rate (percentage).
```

```c
Function:
float calculate_defective_rate(int defective_items, int batch_size);*/
#include<stdio.h>
float defective_rate(int defective,int total)
{
    float rate;
    rate=(defective*100)/total;
    printf("DEfective rate is %f",rate);
}
void main()
{
    int total,defective;
    printf("Enter total and defective number of items\n");
    scanf("%d %d",&total,&defective);
    defective_rate(defective,total);
}
```

```
PS D:\projects\quest\C> cd "d:\proj
Enter total and defective number of
500 20
DEfective rate is 4.000000
PS D:\projects\quest\C>
```

```c
/*Input: Two integers: output rate (units/hour) and downtime (minutes).
Output: Efficiency (percentage).
Function:
float calculate_efficiency(int output_rate, int downtime);*/
#include<stdio.h>
float cal(int or,int dt)
{
    int tt=60,pt;
    pt=tt-dt;
    float efficiency;
    efficiency = (or*pt)/100;
    printf("Efficiency for running %f",efficiency);
}
void main()
{
```

```
    int or,dt;
    printf("Enter output rate and downtime\n");
    scanf("%d %d",&or,&dt);
    cal(or,dt);
}
```

```
PS D:\projects\quest\C> cd "d:\projects\q
Enter output rate and downtime
20 3
Efficiency for running 11.000000
PS D:\projects\quest\C>
```

```
/*Input: Two floating-point numbers: surface area (m²) and paint coverage
per liter (m²/liter).
Output: Required paint (liters).
Function:
float calculate_paint(float area, float coverage);*/
#include<stdio.h>
float paint(float sa,float pc)
{
    float rp;
    rp=sa/pc;
    printf("Required paint %f \n",rp);
}
void main(){
    float sa,pc;
    printf("Enter the surface area and paint coverage per litre\n");
    scanf("%f %f",&sa,&pc);
    paint(sa,pc);
}
```

```
PS D:\projects\quest\C> cd "d:\proje
Enter the surface area and paint cov
200 25
Required paint 8.000000
PS D:\projects\quest\C>
```

```c
/*Input: Two integers: current usage (hours) and maintenance interval
(hours).
Output: Hours remaining for maintenance.
Function:
int calculate_maintenance_schedule(int current_usage, int interval);*/
#include<stdio.h>
int schedule(int cu,int mi)
{
    int i;
    i=cu-mi;
    printf("Interval %d",i);
}
void main()
{
    int cu,mi;
    printf("Enter current usage and maintenance interval\n");
    scanf("%d %d",&cu,&mi);
    schedule(cu,mi);
}
```

```
PS D:\projects\quest\C> cd "d:\projects\ques
Enter current usage and maintenance interval
50 10
Interval 40
PS D:\projects\quest\C>
```

```c
/*Input: Two integers: machine speed (units/hour) and number of operations
per cycle.
Output: Optimal cycle time (seconds).
```

```c
Function:
float calculate_cycle_time(int speed, int operations);*/
#include<stdio.h>
float cycle_time(int ms,int cycle)
{
    float oc;
    oc=ms/cycle;
    printf("Optimal cycle time is %f",oc);
}
void main()
{
    int ms,cycle;
    printf("Enter machine speed and no: cycle\n");
    scanf("%d %d",&ms,&cycle);
    cycle_time(ms,cycle);
}
```

```
PS D:\projects\quest\C> cd "d:\projects
Enter machine speed and no: cycle
1000 20
Optimal cycle time is 50.000000
PS D:\projects\quest\C>
```

```c
/*Write a function that takes the original price of an item and a discount
percentage as parameters.
The function should return the discounted price without modifying the
original price.
Function Prototype:
void calculateDiscount(float originalPrice, float discountPercentage);
 */
#include<stdio.h>
void discount(float op,float d)
{
    float dp;
    dp=op-((op*d)/100);
     printf("Discounted price is %f",dp);
}
void main()
```

```
{
    float op,d;
    printf("Enter price and discount percentage\n");
    scanf("%f %f",&op,&d);
    discount(op,d);

}
```

```
PS D:\projects\quest\C> cd "d:\proje
Enter price and discount percentage
100 34
Discounted price is 66.000000
PS D:\projects\quest\C>
```

```
/*Create a function that takes the current inventory count of a product
and a quantity to add or remove.
The function should return the new inventory count without changing the
original count.
Function Prototype:
int updateInventory(int currentCount, int changeQuantity);
 */
#include<stdio.h>
int update(int stock,int change)
{
    printf("The new stock is %d",stock+change);
}
void main()
{
int stock,change;
printf("Enter current stock and change\n");
scanf("%d %d",&stock,&change);
update(stock,change);

}
```

```
PS D:\projects\quest\C> cd "d:\pr
Enter current stock and change
100 -25
The new stock is 75
PS D:\projects\quest\C>
```

```c
/*Implement a function that accepts the price of an item and a sales tax
rate.
The function should return the total price after tax without altering the
original price.
Function Prototype:
float calculateTotalPrice(float itemPrice, float taxRate);*/
#include<stdio.h>
float total(float price,float tax)
{
    float tt;
    tt=price+(price*tax)/100;
    printf("The total price is %f",tt);
}
void main(){
    float price,tax;
    printf("Enter price and tax\n");
    scanf("%f %f",&price,&tax);
    total(price,tax);
}
```

```
PS D:\projects\quest\C> cd "d:\projects\que
Enter price and tax
300 34
The total price is 402.000000
PS D:\projects\quest\C>
```

```c
/*Design a function that takes the amount spent by a customer and returns
the loyalty points earned based on a
specific conversion rate (e.g., 1 point for every $10 spent). The original
amount spent should remain unchanged.
Function Prototype:
```

```c
int calculateLoyaltyPoints(float amountSpent);*/
#include<stdio.h>
int points(float m)
{
    printf("Points acquired is %f",m/10);
}
void main()
{
    float m;
    printf("Enter amount spent\n");
    scanf("%f",&m);
    points(m);
}
```

```
PS D:\projects\quest\C> cd "d:\project
Enter amount spent
570
Points acquired is 57.000000
PS D:\projects\quest\C>
```

```c
/*Write a function that receives an array of item prices and the number of
items.
 The function should return the total cost of the order without modifying
the individual item prices.
Function Prototype:
float calculateOrderTotal(float prices[], int numberOfItems);*/
#include<stdio.h>
float total(float a[20],int n)
{
    float sum=0;
    for(int i=0;i<n;i++)
    {
        sum+=a[i];
    }
    printf("The total price is %f",sum);
}
void main()
{
    float a[20];
```

```
    int n;
    printf("Enter the number of items\n");
    scanf("%d",&n);
    for(int i=0;i<n;i++)
    {
        scanf("%f",&a[i]);
    }
    total(a,n);
}
```

```
PS D:\projects\quest\C> cd "d:\projects
Enter the number of items
4
10 15 20 24
The total price is 69.000000
PS D:\projects\quest\C>
```

```
/*Create a function that takes an item's price and a refund percentage as
input.
The function should return the refund amount without changing the original
item's price.
Function Prototype:
float calculateRefund(float itemPrice, float refundPercentage);*/
#include<stdio.h>
float refund(float price,float rp)
{
    float r;
    r=(price*rp)/100;
    printf("Refund amount is %f",r);
}
void main()
{
    float price,rp;
    printf("Enter the price and refund percentage\n");
    scanf("%f %f",&price,&rp);
    refund(price,rp);
}
```

```
PS D:\projects\quest\C> cd "d:\projects\quest\C
Enter the price and refund percentage
100 18
Refund amount is 18.000000
PS D:\projects\quest\C>
```

```c
/*Implement a function that takes the weight of a package and calculates
shipping costs
based on weight brackets (e.g., $5 for up to 5kg, $10 for 5-10kg). The
original weight should remain unchanged.
Function Prototype:
float calculateShippingCost(float weight);*/
#include<stdio.h>
float shipping(float w)
{
    if(w<5)
    printf("Shipping cost is $5\n");
    else if(5<=w<10)
    printf("Shipping cost id $10\n");
}
void main()
{
    float weight;
    printf("Enter weight\n");
    scanf("%f",&weight);
    shipping(weight);
}
```

```
PS D:\projects\quest\C> cd "d:\projects\quest\C
Enter weight
60
Shipping cost id $10
PS D:\projects\quest\C>
```

```c
/*Design a function that converts an amount from one currency to another
based on an exchange rate provided as input.
The original amount should not be altered.
Function Prototype:
float convertCurrency(float amount, float exchangeRate);
 */

#include<stdio.h>
float exchange(float amount,float rate)
{
    printf("Converted amount is %f",amount*rate);
}
void main()
{
    float amount,rate;
    printf("Enter the amount and rate\n");
    scanf("%f %f",&amount,&rate);
    exchange(amount,rate);
}
```

```
PS D:\projects\quest\C> cd "d:\projects\quest\
Enter the amount and rate
100 16
Converted amount is 1600.000000
PS D:\projects\quest\C>
```

```c
/*Write a function that takes two prices from different vendors and
returns the lower price without modifying either input price.
Function Prototype:
float findLowerPrice(float priceA, float priceB);*/
```

```c
#include<stdio.h>
float lower(float p1,float p2)
{
    if(p1>p2)
    printf("%f is the lower amount\n",p2);
    else
    printf("%f is the lower amount",p1);
}
void main()
{
    float p1,p2;
    printf("Enter the two prices\n");
    scanf("%f %f",&p1,&p2);
    lower(p1,p2);
}
```

```
PS D:\projects\quest\C> cd "d:\pr
Enter the two prices
40 22
22.000000 is the lower amount
PS D:\projects\quest\C>
```

```c
/*Create a function that checks if a customer is eligible for a senior
citizen discount based on their age.
 The function should take age as input and return whether they qualify
without changing the age value.
Function Prototype:
bool isEligibleForSeniorDiscount(int age);
*/
#include<stdio.h>
#include<stdbool.h>
bool senior(float age)
{
    if(age>50)
    printf("Eligible for senior discount\n");
    else
    printf("Not eligible for senior discount\n");
}
```

```c
void main()
{
    int age;
    printf("Enter the age\n");
    scanf("%d",&age);
    senior(age);
}
```

```
PS D:\projects\quest\C> cd "d:\projects\ques
Enter the age
55
Eligible for senior discount
PS D:\projects\quest\C>
```