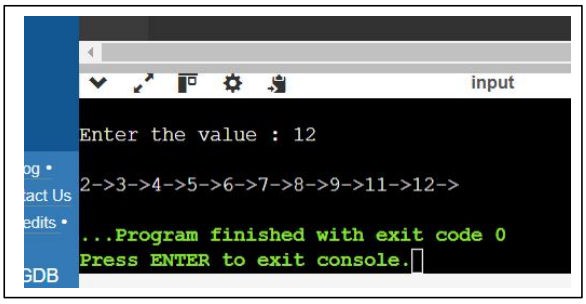
ASSIGNMENT

Variable Length Array

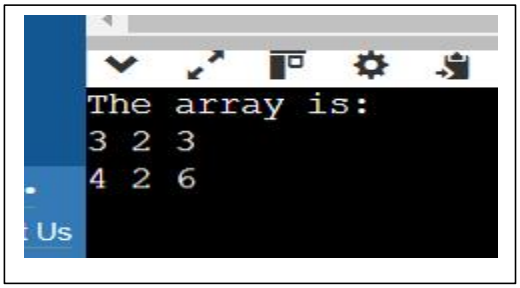
One Dimensional Array

```
#include<stdio.h>
int main(){

   int n=10;
   int ar[n];
   for(int i=0;i<10;i++){
      printf("Enter the value : ");
      scanf("%d",&ar[i]);
      printf("\n");
   }
   for(int j=0;j<10;j++){
      printf("%d->",ar[j]);

   }
   return 0;
}
```



Two Dimensional Array

```
#include <stdio.h>
int main() {
   int m = 2;
   int n = 3;
   int ar[m][n];

   for(int i = 0; i < m; i++) {
      for(int j = 0; j < n; j++) {
         printf("Enter the value for ar[%d][%d]: ", i, j);
         scanf("%d", &ar[i][j]);
      }
      printf("\n");
   }

   printf("The array is:\n");
   for(int i = 0; i < m; i++) {
      for(int j = 0; j < n; j++) {
         printf("%d ", ar[i][j]);
      }
      printf("\n");
   }
```

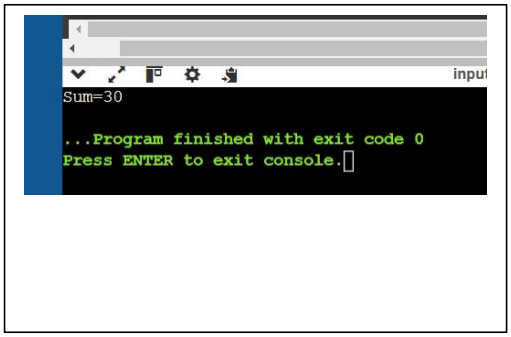


```
    return 0;
}
```

Functions

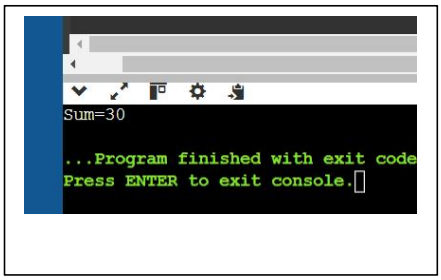
Write a program to add two number using the add function without passing any parameter and the function is not going to return any data

```
#include<stdio.h>
void add_num(void);
int main(){
    add_num();
    return 0;
}
void add_num(){
    int a =10,b=20,sum=0;
    sum=a+b;
    printf("Sum=%d",sum);
}
```



Write a program to add two number using the add function

```
#include<stdio.h>
void add_num(int ,int);
int main(){
    int a=10,b=20;
    add_num(a,b);
    return 0;
}
void add_num(int a,int b){
    int sum=0;
    sum=a+b;
    printf("Sum=%d",sum);
}
```



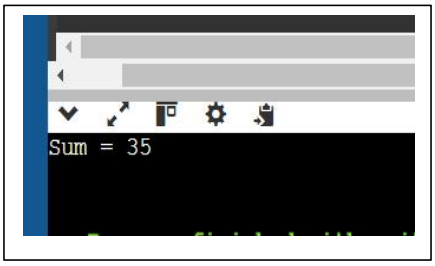
Write a program on functions

```
#include <stdio.h>

void add_num(int , int);

int main(){
    int a = 10, b = 20;
    printf("001a = %p\n",&a);
    printf("001b = %p\n",&b);
    add_num(a,b);

    printf("The values of a and b is %d , %d",a,b);
```

```
    return 0;
}

void add_num(int a, int b){
    a = 40;
    b = 50;
    printf("002a = %p\n",&a);
    printf("002b = %p\n",&b);
    int sum =0;
    sum = a + b;
    printf("Sum = %d \n",sum);
}
```

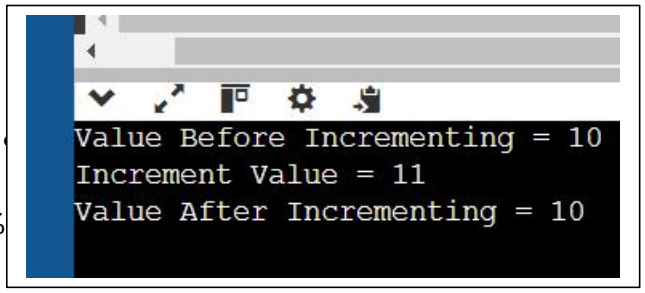
Write a program to add two number using the add function by parameter and the function is going to return any Data

```
#include <stdio.h>
int add_num(int,int);
int main(){
    int a = 15, b = 20;
    int sum=0;
    sum=add_num(a,b);
    printf("Sum = %d \n",sum);
    return 0;
}

int add_num(int a, int b){
    int sum=a+b;
    return sum;
}
```

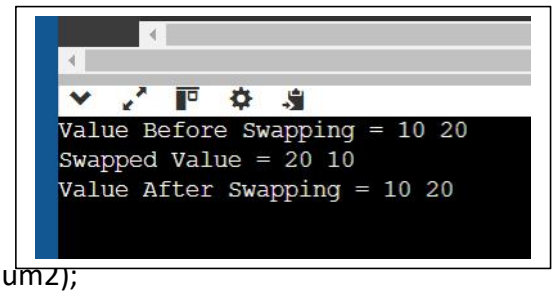


Create a C program that defines a function to increment an integer by 1. The function should demonstrate call by value, showing that the original value remains unchanged.

```
#include<stdio.h>
void increment_num(int num);
int main(){
    int num=10;
    printf("Value Before Incrementing = 
    increment_num(num);
    printf("Value After Incrementing = %
    return 0;
}
void increment_num(int num){
    num=num+1;
    printf("Increment Value = %d\n",num);
}
```

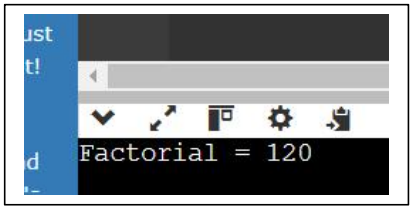


Write a C program that attempts to swap two integers using a function that employs call by value. Show that the original values remain unchanged after the function call.



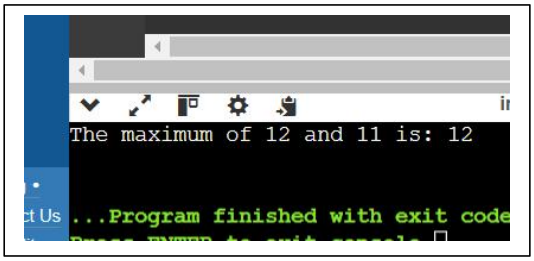

```
#include<stdio.h>
void swapping_num(int num1,int num2);
int main(){
   int num1=10,num2=20;
   printf("Value Before Swapping = %d %d\n",num1,num2);
   swapping_num(num1,num2);
   printf("Value After Swapping = %d %d\n",num1,num2);
   return 0;
}
void swapping_num(int num1,int num2){
   num1=num1+num2;
   num2=num1-num2;
   num1=num1-num2;
   printf("Swapped Value = %d %d\n",num1,num2);
}
```

Develop a C program that calculates the factorial of a number using call by value.




```
#include<stdio.h>
void factorial(int num);
int main(){
   int num=5;
   factorial(num);
   return 0;
}
void factorial(int num){
   int fact=1;
   for(int i=2;i<=num;i++){
      fact*=i;
   }
   printf("Factorial = %d\n",fact);
}
```

Create a C program that defines a function to find the maximum of two numbers using call by value




```
#include <stdio.h>

int maximum(int a, int b);

int main() {
   int num1 = 12, num2 = 11;
   int max;
```

```
    max = maximum(num1, num2);

    printf("The maximum of %d and %d is: %d\n", num1, num2, max);

    return 0;
}

int maximum(int num1, int num2) {
    if (num1 > num2) {
        return num1;
    } else {
        return num2;
    }
}
```

Problem Statement 1: Arithmetic Operations Calculator

Description: Write a C program that performs basic arithmetic operations (addition, subtraction, multiplication, and division) on two numbers provided by the user. The program should use functions to perform each operation and demonstrate call by value.

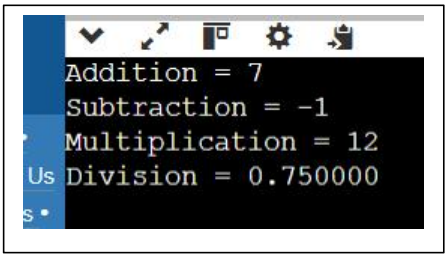
Requirements:
Create separate functions for addition, subtraction, multiplication, and division.
Each function should take two parameters (the numbers) and return the result.
Use appropriate data types for the variables.
Use operators for arithmetic calculations.

Example Input/Output:

Enter first number: 10
Enter second number: 5
Addition: 15
Subtraction: 5
Multiplication: 50
Division: 2.0

```
#include <stdio.h>

int add_num(int,int);
int sub_num(int,int);
int mul_num(int,int);
float div_num(int,int);

int main(){
    int a, b,sum=0,diff=0,product=0;
```

```
    float divi=0.0;

    printf("Enter First Number : ");
    scanf("%d",&a);
    printf("Enter Second Number : ");
    scanf("%d",&b);

    sum=add_num(a,b);
    printf("Addition = %d \n",sum);

    diff=sub_num(a,b);
    printf("Subtraction = %d \n",diff);

    product=mul_num(a,b);
    printf("Multiplication = %d \n",product);

    divi=div_num(a,b);
    printf("Division = %f \n",divi);

    return 0;
}

int add_num(int a, int b){
    int sum=a+b;
    return sum;
}
int sub_num(int a, int b){
    int diff=a-b;
    return diff;
}
int mul_num(int a, int b){
    int product=a*b;
    return product;
}
float div_num(int a, int b){
    if (b == 0) {
        printf("Error: Division by zero is not allowed.\n");
        return 0.0;
        }
        else {
            float divi= (float)a / b;
            return divi;
        }

}
```



Problem Statement 2: Temperature Conversion

Description: Develop a C program that converts temperatures between Celsius and Fahrenheit. The program should use functions to handle the conversions and demonstrate call by value.

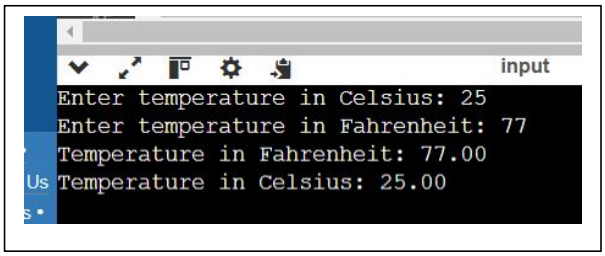
Requirements:
Create two functions: one for converting Celsius to Fahrenheit and another for converting Fahrenheit to Celsius.
Each function should accept a temperature value as an argument and return the converted temperature.
Use appropriate data types for temperature values.
Use arithmetic operators to perform the conversion calculations.

Example Input/Output:

Enter temperature in Celsius: 25
Temperature in Fahrenheit: 77.0

Enter temperature in Fahrenheit: 77
Temperature in Celsius: 25.0

```
#include <stdio.h>

float celsius_to_fahrenheit(int celsius);
float fahrenheit_to_celsius(int fahrenheit);

int main() {
   int celsius, fahrenheit;
   float celsiustofahren, fahrentocelsius;

   printf("Enter temperature in Celsius: ");
   scanf("%d", &celsius);
   printf("Enter temperature in Fahrenheit: ");
   scanf("%d", &fahrenheit);

   celsiustofahren = celsius_to_fahrenheit(celsius);
   printf("Temperature in Fahrenheit: %.2f\n", celsiustofahren);

   fahrentocelsius = fahrenheit_to_celsius(fahrenheit);
   printf("Temperature in Celsius: %.2f\n", fahrentocelsius);

   return 0;
}

float celsius_to_fahrenheit(int celsius) {
   return (celsius * 9.0/5.0) + 32;
}
```



```
float fahrenheit_to_celsius(int fahrenheit) {
   return (fahrenheit - 32) * 5.0/9.0;
}
```

Problem Statement 3: Simple Interest Calculator

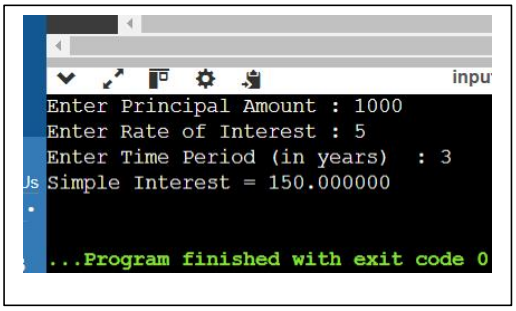
Description: Develop a C program that calculates simple interest based on user input for principal amount, rate of interest, and time period. The program should use a function to compute interest and demonstrate call by value.

Requirements:
Implement a function that takes three parameters (principal, rate, time) and returns the calculated simple interest.
Use appropriate data types for financial calculations (e.g., float or double).
Utilize arithmetic operators to compute simple interest using the formula
SI = P×R×T/100

Example Input/Output:

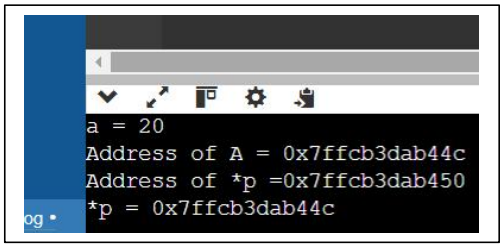
Enter principal amount: 1000
Enter rate of interest: 5
Enter time period (in years): 3
Simple Interest is: 150.0

```
#include <stdio.h>

int simple_interests(int,int,int);

int main(){
   int principal,rate,years;
   float simple_interest;

   printf("Enter Principal Amount : ");
   scanf("%d",&principal);
   printf("Enter Rate of Interest : ");
   scanf("%d",&rate);
   printf("Enter Time Period (in years)  : ");
   scanf("%d",&years);

   simple_interest=simple_interests(principal,rate,years);
   printf("Simple Interest = %f \n",simple_interest);


   return 0;
}
```



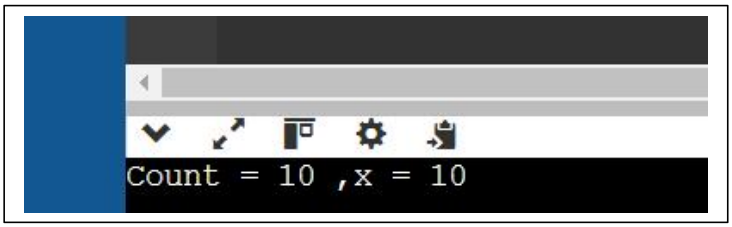
```
int simple_interests(int principal, int rate, int years){
   float simple_interest=(principal*rate*years)/100;
   return simple_interest;
}
```

Pointers




```
#include<stdio.h>

int main(){
   int a;
   int *p;
   p=&a;
   *p=20;

   printf("a = %d \n",a);
   printf("Address of A = %p \n",&a);
   printf("Address of *p =%p \n",&p);
   printf("*p = %p \n",p);

   return 0;
}
```

Calculating Count

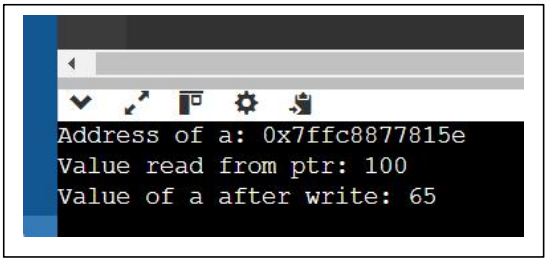



```
#include<stdio.h>

int main(){
   int count=10,x;
   int *pCount=&count;
   x=*pCount;
   printf("Count = %d ,x = %d",count,x);
   return 0;
}
```

Exercise

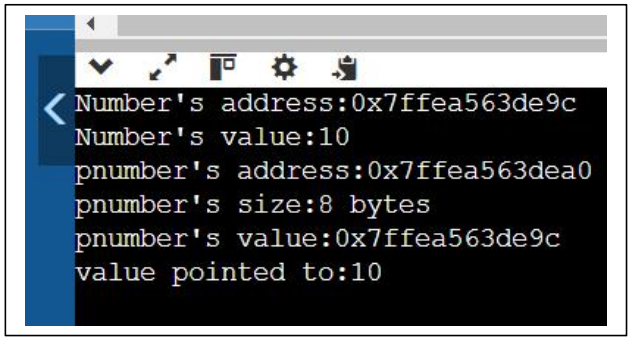
1) Create a char type variable and initialize it to value 100
2) Print the address of the above variable.
3) Create a pointer variable and store the address of the above variable
4) Perform read operation on the pointer variable to fetch 1 byte of data from the pointer
5) Print the data obtained from the read operation on the pointer.
6) Perform write operation on the pointer to store the value 65
7) Print the value of the variable defined in step 1

```
#include <stdio.h>

int main() {
   char a = 100;

   printf("Address of a: %p\n", &a);
   char *ptr = &a;
   char value = *ptr;
   printf("Value read from pointer: %d\n", value);
   *ptr = 65;
   printf("Value of a after write: %d\n", a);

   return 0;
}
```



Example

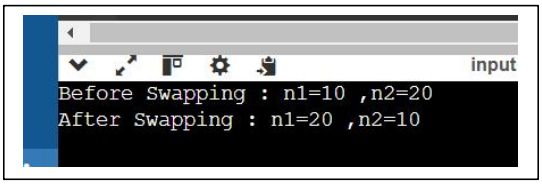
```
#include<stdio.h>

int main(void){
   int number=0;
   int *pnumber=NULL;

   number=10;
   printf("Number's address:%p\n",&number);
   printf("Number's value:%d\n",number);
   pnumber=&number;
   printf("pnumber's address:%p\n",(void*)&pnumber);
   printf("pnumber's size:%zd bytes\n",sizeof(pnumber));
```



```
    printf("pnumber's value:%p\n",pnumber);
    printf("value pointed to:%d\n",*pnumber);

    return 0;
}
```

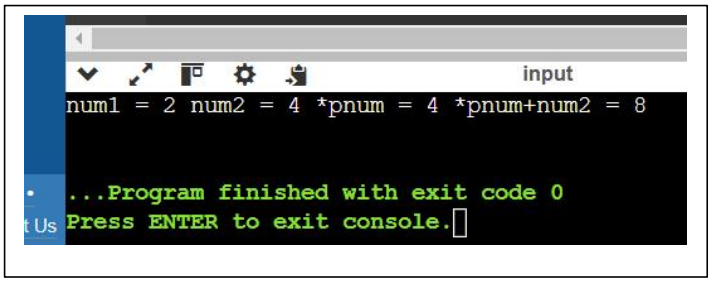
Write a C program that swaps the values of two integers using pointers.

```
#include<stdio.h>

int main(){
    int n1=10,n2=20;
    int *p_n1=&n1;
    int *p_n2=&n2;
    printf("Before Swapping : n1=%d ,n2=%d \n",n1,n2);
    int temp=*p_n1;
    *p_n1=*p_n2;
    *p_n2=temp;
       printf("After Swapping : n1=%d ,n2=%d \n",n1,n2);
    return 0;

}
```



Example

```
#include<stdio.h>

int main(){
    long num1=0;
    long num2=0;
    long *pnum=NULL;

    pnum=&num1;
    *pnum=2;
    ++num2;
    num2+=*pnum;
    pnum=&num2;
    ++*pnum;
    printf("num1 = %ld num2 = %ld *pnum = %ld *pnum+num2 = %ld
\n",num1,num2,*pnum,*pnum+num2);
    return 0;

}
```

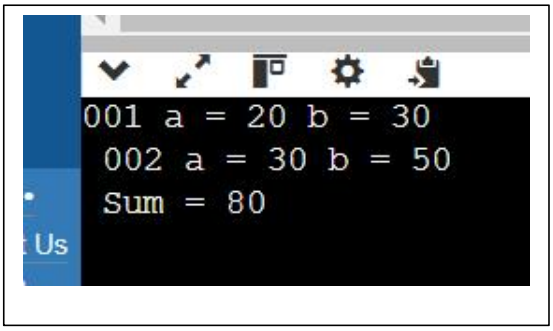


Testing Example

```
#include<stdio.h>

int main(){
   int a;
   int *p;
   if(p !=0){
      *p=5;
   }
   printf("a = %d",a);
   return 0;

}
```

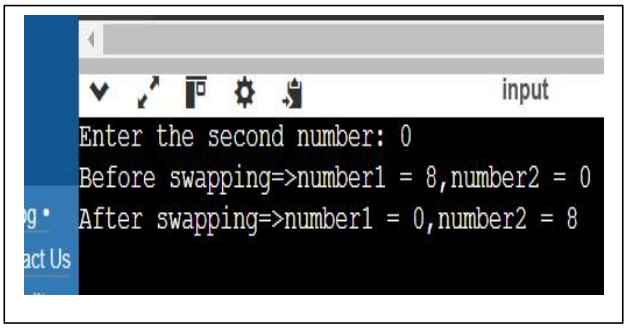
Pass by reference

```
#include<stdio.h>

int addnum(int *,int *);

int main(){

   int a=20,b=30;
   printf("001 a = %d b = %d\n ",a,b);
   int sum=addnum(&a,&b);
   printf("002 a = %d b = %d\n ",a,b);
   printf("Sum = %d \n",sum);

   return 0;

}
int addnum(int *p,int *q){
   *p=30;
   *q=50;
   int s=*p + *q;
   return s;
}
```



Write a program to swap the number using swap function and follow the pass by reference method.

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

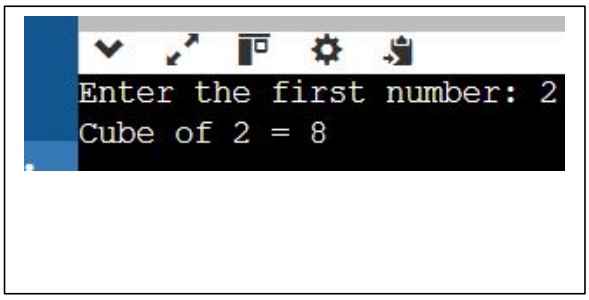



```
void swap_num(int *num1, int *num2);

int main() {

   int number1, number2;

   printf("Enter the first number: ");

   scanf("%d", &number1);

   printf("Enter the second number: ");

   scanf("%d", &number2);

   printf("Before swapping=>number1 = %d,number2 = %d\n", number1, number2);

   swap_num(&number1, &number2);

   printf("After swapping=>number1 = %d,number2 = %d\n", number1, number2);

   return 0;

}

void swap_num(int *num1, int *num2) {

   int temp;

   temp = *num1;

   *num1 = *num2;

   *num2 = temp;

}
```

Write for Finding the Cube of a Number Using Pass by Reference

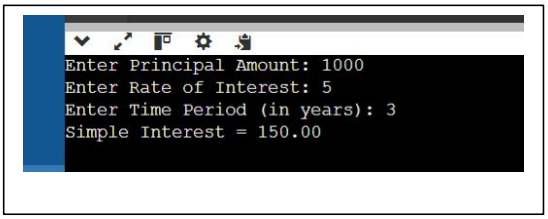
```
#include <stdio.h>


int cube_num(int *n);


int main() {
```

```
    int num;

    printf("Enter the first number: ");
    scanf("%d", &num);

    int cube = cube_num(&num);

    printf("Cube of %d = %d\n", num, cube);

    return 0;
}

int cube_num(int *n) {
    int c = (*n) * (*n) * (*n);
    return c;
}
```



Write a program to calculate the simple interest with the help of a function and pass call by reference method.

```
#include <stdio.h>

void calculate_simple_interest(int principal, int rate, int years, float
*simple_interest);

int main() {
    int principal, rate, years;
    float simple_interest;
```




```
    printf("Enter Principal Amount: ");
    scanf("%d", &principal);
    printf("Enter Rate of Interest: ");
    scanf("%d", &rate);
    printf("Enter Time Period (in years): ");
    scanf("%d", &years);

    calculate_simple_interest(principal, rate, years, &simple_interest);

    printf("Simple Interest = %.2f\n", simple_interest);

    return 0;
}

void calculate_simple_interest(int principal, int rate, int years, float *simple_interest)
{
    *simple_interest = (principal * rate * years) / 100.0;
}
```