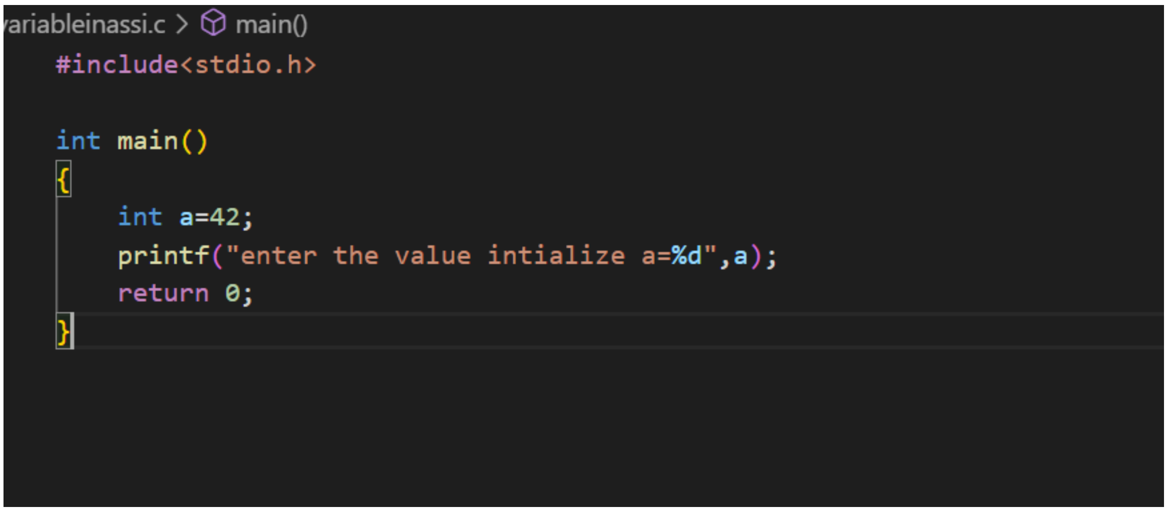
ANSU MARIUM SHIBU

DAY 3-ASSIGNMENTS

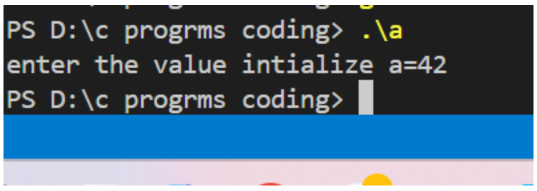
1.Variable Initialization

Question: Write a program that declares an integer variable, initializes it with a value of 42, and prints the value to the console.



```
variableinassi.c > main()
#include<stdio.h>

int main()
{
    int a=42;
    printf("enter the value intialize a=%d",a);
    return 0;
}
```

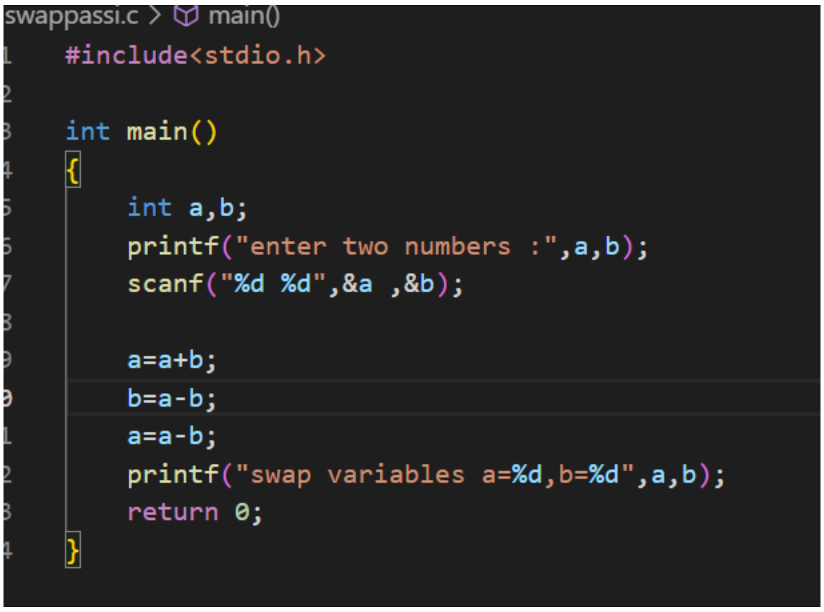
Output:



```
PS D:\c progrms coding> .\a
enter the value intialize a=42
PS D:\c progrms coding>
```

2. Swapping Variables

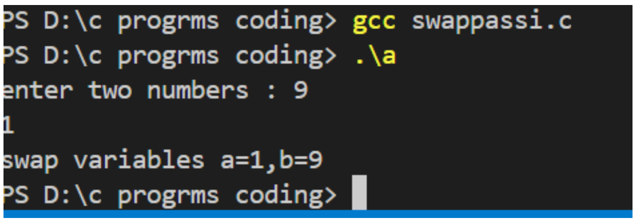
Question: Create a program that swaps the values of two integer variables without using a temporary variable. Demonstrate this by printing the values before and after the swap.



swappassi.c > main()

```c
#include<stdio.h>

int main()
{
    int a,b;
    printf("enter two numbers :",a,b);
    scanf("%d %d",&a ,&b);

    a=a+b;
    b=a-b;
    a=a-b;
    printf("swap variables a=%d,b=%d",a,b);
    return 0;
}
```

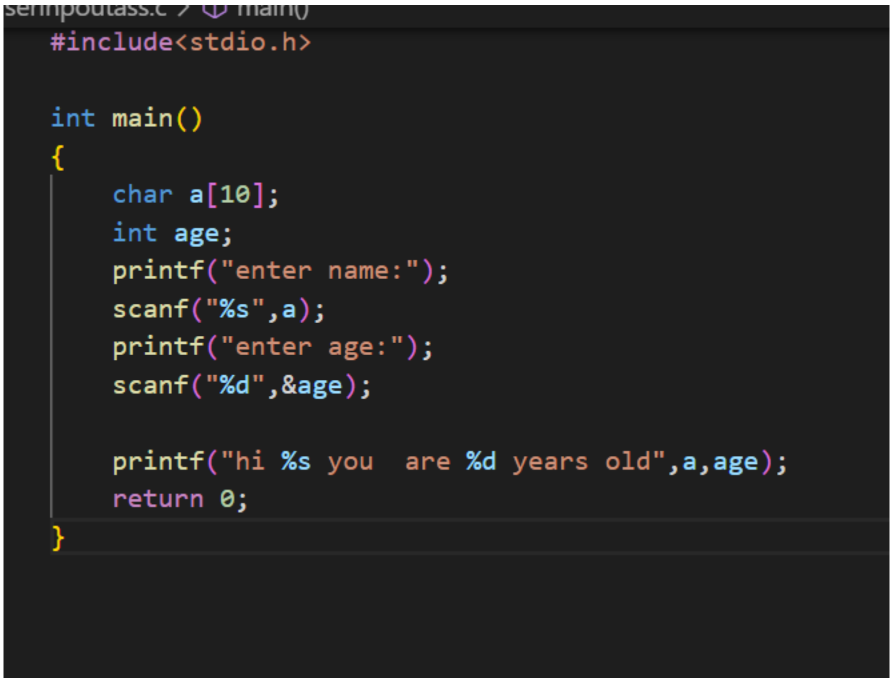
Output:



```
PS D:\c progrms coding> gcc swappassi.c
PS D:\c progrms coding> .\a
enter two numbers : 9
1
swap variables a=1,b=9
PS D:\c progrms coding>
```

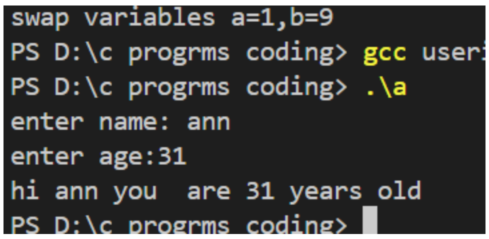
3. User Input and Output

Question: Write a program that prompts the user to enter their name and age, stores these values in appropriate variables, and then prints a greeting message that includes both the name and age.



```c
#include<stdio.h>

int main()
{
    char a[10];
    int age;
    printf("enter name:");
    scanf("%s",a);
    printf("enter age:");
    scanf("%d",&age);

    printf("hi %s you  are %d years old",a,age);
    return 0;
}
```

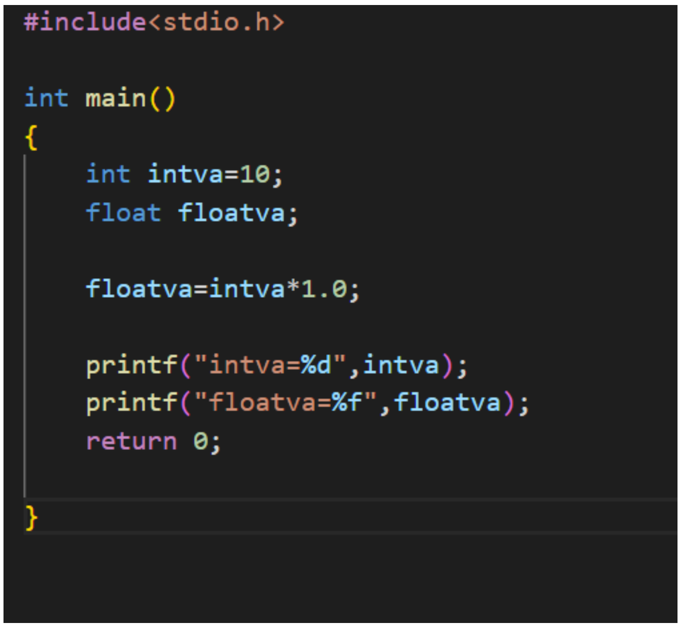
Output:



```
swap variables a=1,b=9
PS D:\c progrms coding> gcc user
PS D:\c progrms coding> .\a
enter name: ann
enter age:31
hi ann you  are 31 years old
PS D:\c progrms coding>
```

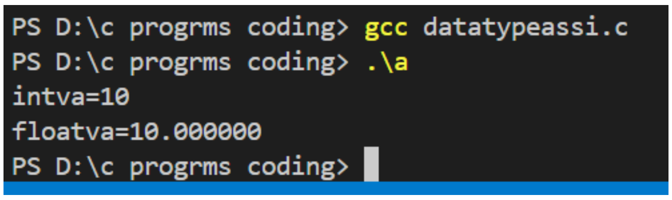
4. Data Type Conversion

Question: Write a program that declares an integer variable, assigns it a value of 10, and then converts it to a float variable. Print both the integer and float values to show the conversion.



```c
#include<stdio.h>

int main()
{
    int intva=10;
    float floatva;

    floatva=intva*1.0;

    printf("intva=%d",intva);
    printf("floatva=%f",floatva);
    return 0;

}
```

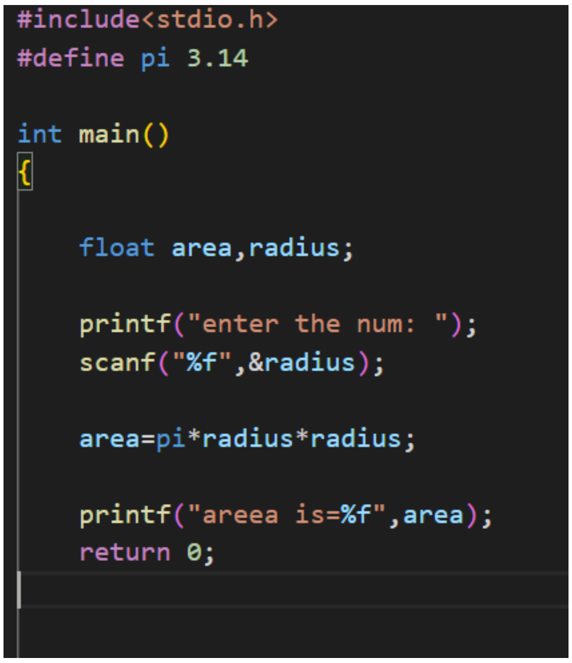
Output:



```
PS D:\c progrms coding> gcc datatypeassi.c
PS D:\c progrms coding> .\a
intva=10
floatva=10.000000
PS D:\c progrms coding> 
```

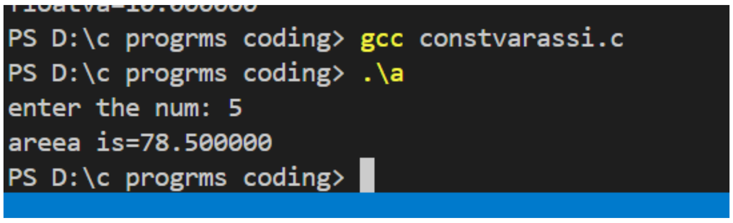
5. Constants vs. Variables

Question: Using #define, create a constant for the value of Pi (3.14). Write a program that calculates the area of a circle given its radius (stored in a variable) and prints the result using the constant for Pi.



```c
#include<stdio.h>
#define pi 3.14

int main()
{

    float area,radius;

    printf("enter the num: ");
    scanf("%f",&radius);

    area=pi*radius*radius;

    printf("areea is=%f",area);
    return 0;

```

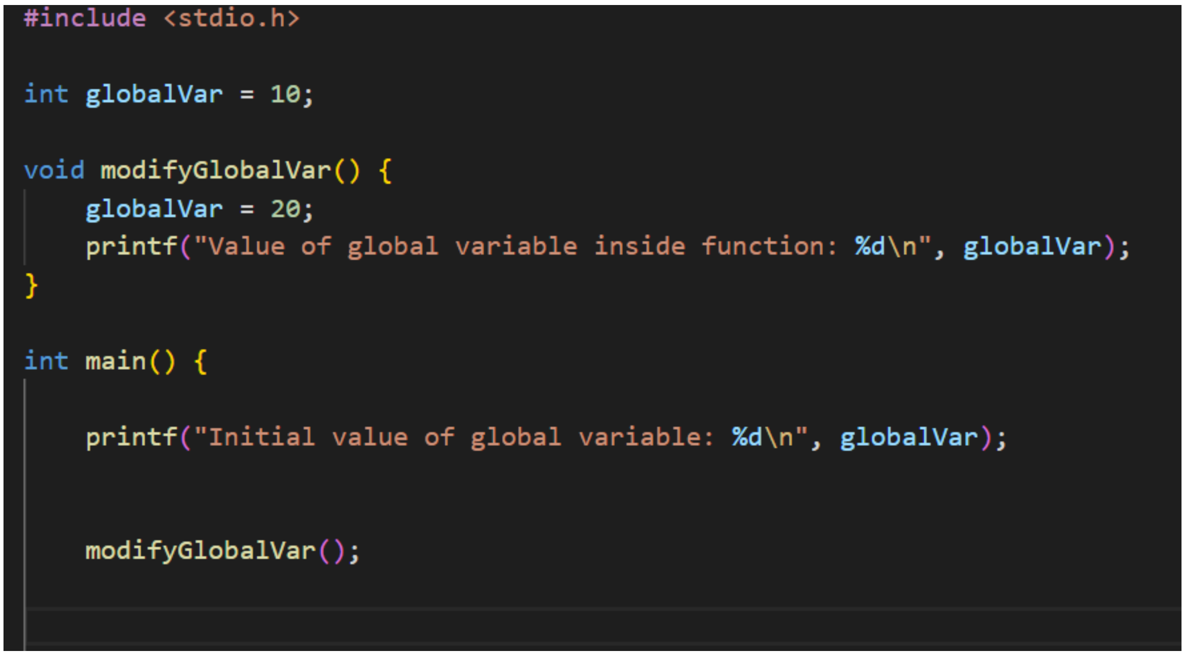
Output:



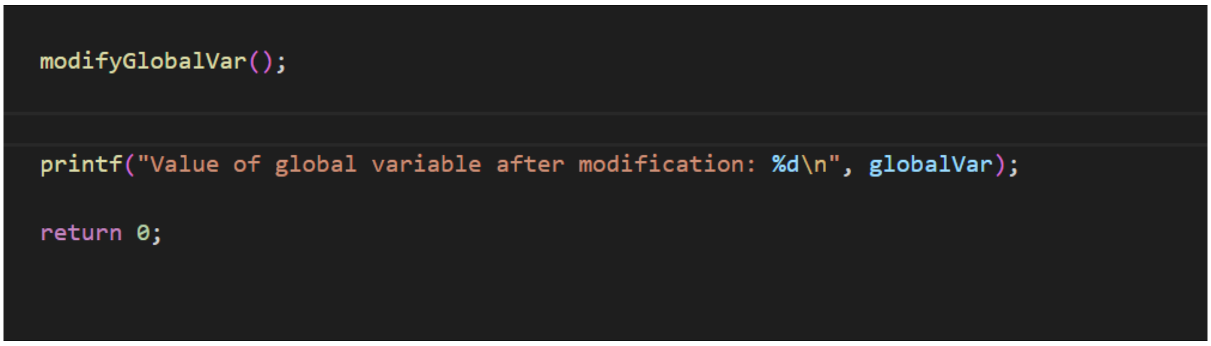
```
floatva=10.000000
PS D:\c progrms coding> gcc constvarassi.c
PS D:\c progrms coding> .\a
enter the num: 5
areea is=78.500000
PS D:\c progrms coding> 
```

6. Scope of Variables

Question: Write a program that demonstrates the concept of variable scope by declaring a global variable and modifying it within a function. Print the value of the global variable before and after modification.

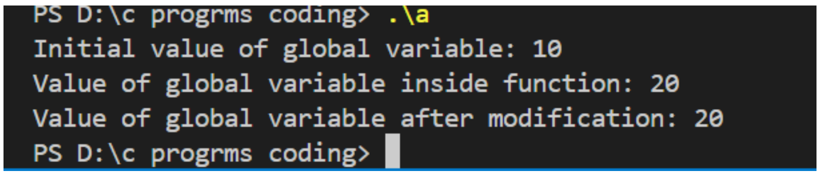


```c
#include <stdio.h>

int globalVar = 10;

void modifyGlobalVar() {
    globalVar = 20;
    printf("Value of global variable inside function: %d\n", globalVar);
}

int main() {

    printf("Initial value of global variable: %d\n", globalVar);


    modifyGlobalVar();
```



```c
    modifyGlobalVar();


    printf("Value of global variable after modification: %d\n", globalVar);

    return 0;
```

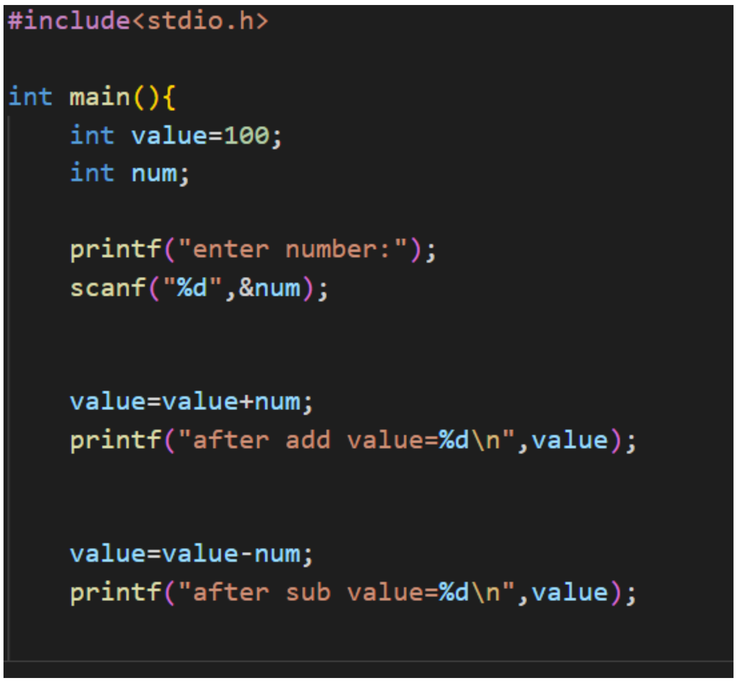
Output:



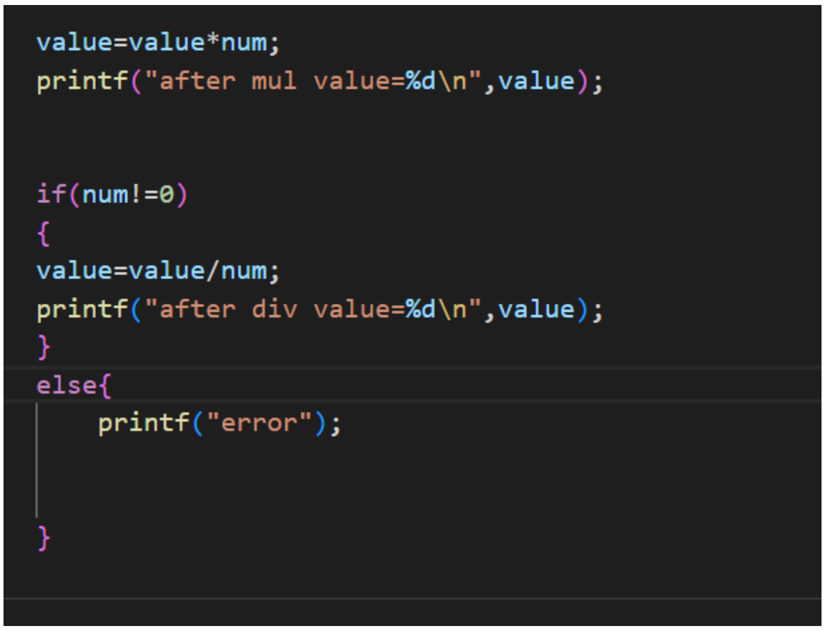
```
PS D:\c progrms coding> .\a
Initial value of global variable: 10
Value of global variable inside function: 20
Value of global variable after modification: 20
PS D:\c progrms coding> 
```

8. Using Augmented Assignment Operators

Question: Write a program that uses augmented assignment operators (+=, -=, *=, /=) to perform calculations on an integer variable initialized to 100. Print the value after each operation.

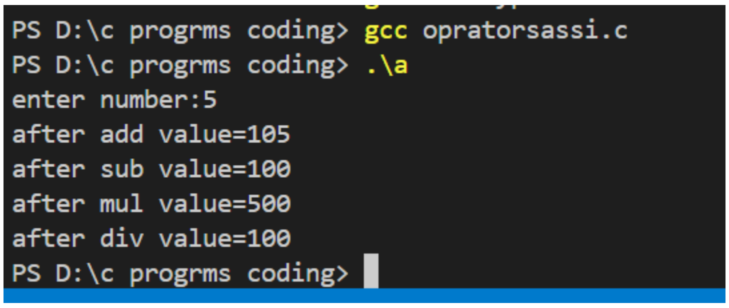


```c
#include<stdio.h>

int main(){
    int value=100;
    int num;

    printf("enter number:");
    scanf("%d",&num);


    value=value+num;
    printf("after add value=%d\n",value);


    value=value-num;
    printf("after sub value=%d\n",value);

```



```c
  value=value*num;
  printf("after mul value=%d\n",value);


  if(num!=0)
  {
  value=value/num;
  printf("after div value=%d\n",value);
  }
  else{
      printf("error");


  }
```

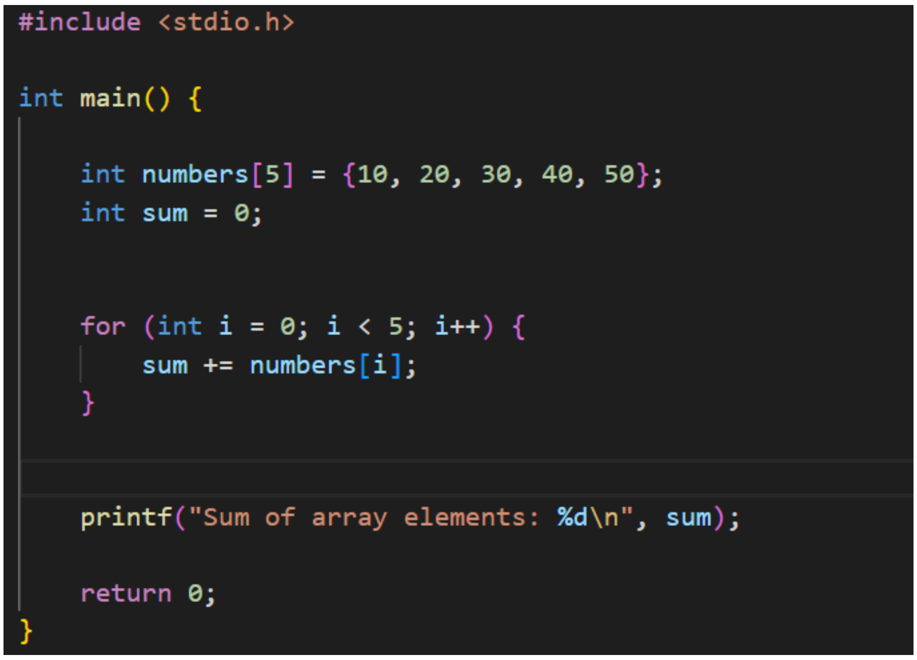
Output:



```
PS D:\c progrms coding> gcc opratorsassi.c
PS D:\c progrms coding> .\a
enter number:5
after add value=105
after sub value=100
after mul value=500
after div value=100
PS D:\c progrms coding>
```

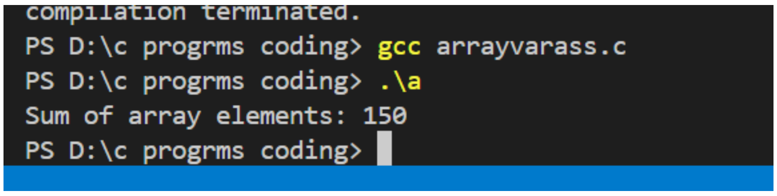
9.Array of Variables

Question: Create an array of integers with five elements. Initialize it with values of your choice, then write a program to calculate and print the sum of all elements in the array.



```c
#include <stdio.h>

int main() {

    int numbers[5] = {10, 20, 30, 40, 50};
    int sum = 0;


    for (int i = 0; i < 5; i++) {
        sum += numbers[i];
    }



    printf("Sum of array elements: %d\n", sum);

    return 0;
}
```

Output:



```
compilation terminated.
PS D:\c progrms coding> gcc arrayvarass.c
PS D:\c progrms coding> .\a
Sum of array elements: 150
PS D:\c progrms coding>
```

10. Assignment: User Authentication Program

Objective

Create a C program that prompts the user for a username and password, then checks if the entered credentials match predefined values. Use logical operators to determine if the authentication is successful.

Requirements

Define two constants for the correct username and password.

Prompt the user to enter their username and password.

Use logical operators (&&, ||, !) to check if:

If both are correct, display a success message.

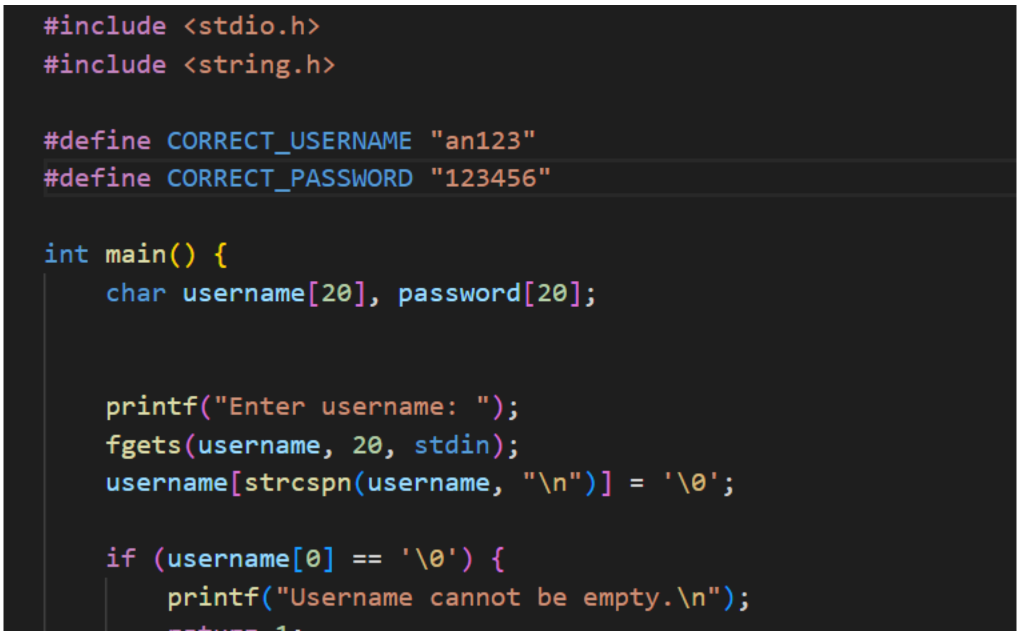
Implement additional checks:

If the username is empty, display a message indicating that the username cannot be empty.

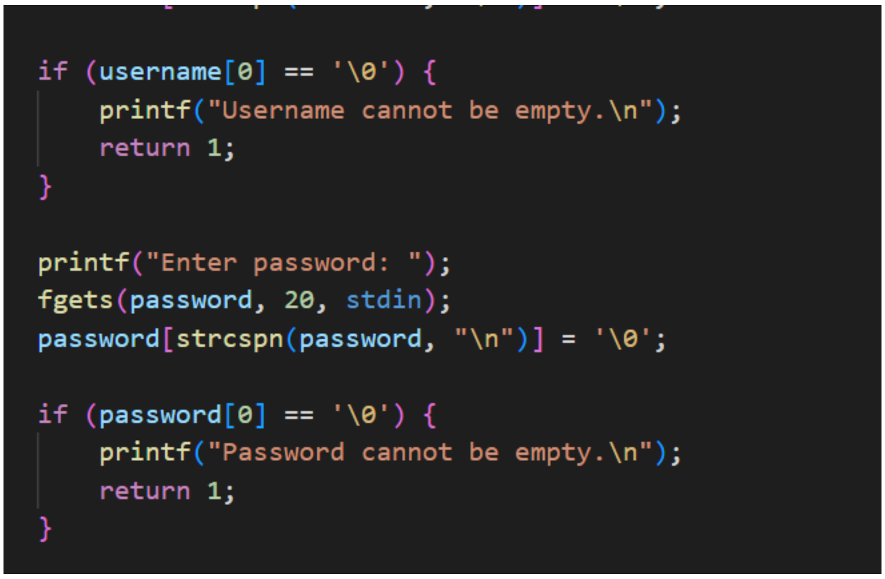
If the password is empty, display a message indicating that the password cannot be empty.

The username matches the predefined username AND the password matches the predefined password.

If either the username or password is incorrect, display an appropriate error message.

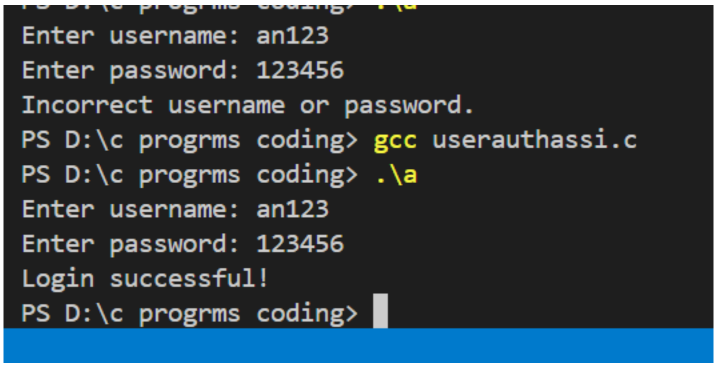


```c
#include <stdio.h>
#include <string.h>

#define CORRECT_USERNAME "an123"
#define CORRECT_PASSWORD "123456"

int main() {
    char username[20], password[20];


    printf("Enter username: ");
    fgets(username, 20, stdin);
    username[strcspn(username, "\n")] = '\0';

    if (username[0] == '\0') {
        printf("Username cannot be empty.\n");
```



```c
if (username[0] == '\0') {
    printf("Username cannot be empty.\n");
    return 1;
}

printf("Enter password: ");
fgets(password, 20, stdin);
password[strcspn(password, "\n")] = '\0';

if (password[0] == '\0') {
    printf("Password cannot be empty.\n");
    return 1;
}
```

Output:



```
Enter username: an123
Enter password: 123456
Incorrect username or password.
PS D:\c progrms coding> gcc userauthassi.c
PS D:\c progrms coding> .\a
Enter username: an123
Enter password: 123456
Login successful!
PS D:\c progrms coding> 
```