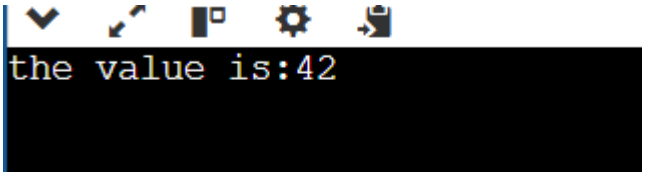
1. Variable Initialization

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

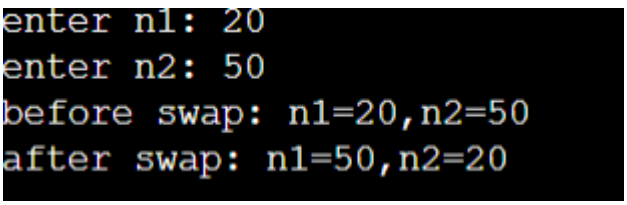
```
#include<stdio.h>
int main(){
int n=42;
printf("the value is:%d\n",n);
return 0;
}
```



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

```
#include<stdio.h>
int main(){
int n1,n2;
printf("enter n1: ");
scanf("%d",&n1);
printf("enter n2: ");
scanf("%d",&n2);
printf("before swap: n1=%d,n2=%d\n",n1,n2);
n1=n1+n2;
n2=n1-n2;
n1=n1-n2;
printf("after swap: n1=%d,n2=%d\n",n1,n2);
return 0;
}
```



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 n

```
#include<stdio.h>
int main(){
int age;
char name[30];
printf("enter name: ");
scanf("%s", &name);
printf("enter age: ");
scanf("%d",&age);
printf("hello %s, you are %d years old\n",name,age);
return 0;
}
```

```
enter name: vaisakh
enter age: 21
hello vaisakh, you are 21 years old
```

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.

```
#include<stdio.h>
int main(){
int n=42;
float f1;
f1=(float) n;
printf("the integer value is:%d\n",n);
printf("the float value is:%.2f\n",f1);
return 0;
}
```

```
the integer value is:42
the float value is:42.00
```

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.

```
#include<stdio.h>
#define Pi 3.14
int main(){
float r,a;
printf("enter radius of circle: ");
scanf("%f",&r);
a=Pi*r*r;
printf("the area of circle is :%.2f\n",a);
return 0;
}
```

```
enter radius of circle: 3
the area of circle is :28.26
```

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

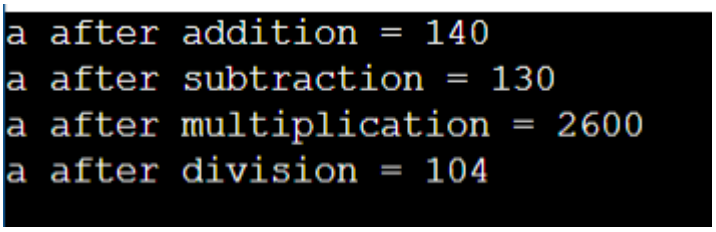
```
#include <stdio.h>
int globalvar = 30;
int mod() {
globalvar = 20;
printf("Value of globalvar inside function: %d\n", globalvar);
}
int main() {
printf("Value of globalvar before modification: %d\n", globalvar);
mod();
printf("Value of globalvar after modification: %d\n", globalvar);
return 0;
}
```

```
Value of globalvar before modification: 30
Value of globalvar inside function: 20
Value of globalvar after modification: 20
```

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.

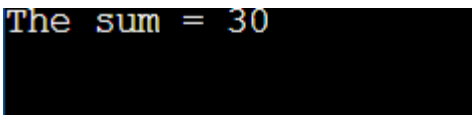
```
#include<stdio.h>
int main() {
int a = 100;
a += 40;
printf("a after addition = %d\n", a);
a -= 10;
printf("a after subtraction = %d\n", a);
a *= 20;
printf("a after multiplication = %d\n", a);
a /= 25;
printf("a after division = %d\n", a);
return 0;
}
```



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.

```
#include <stdio.h>
int main() {
int a[3] = {5, 10, 15};
int sum = 0;
for (int i = 0; i < 3; i++) {
sum += a[i];
}
printf("The sum = %d\n", sum);
return 0;
}
```



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

1. Define two constants for the correct username and password.
2. Prompt the user to enter their username and password.
3. Use logical operators (`&&`, `||`, `!`) to check if:
4. If both are correct, display a success message.
5. 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.

```
#include<stdio.h>
#include<string.h>
#define username "Window"
#define password "12345"
int main()
{
char myusername[20];
char mypassword[20];
printf("enter username: \n");
scanf("%s",&myusername);
printf("enter password: \n");
scanf("%s",&mypassword);
if(strlen(myusername)==0){
printf("username cannot be empty");
}
if(strlen(mypassword)==0){
printf("password cannot be empty");
}
{
if(strcmp(myusername,username)==0 &&
strcmp(mypassword,password)==0)
{
```

```
printf("authentication successfull");
}else{
printf("error");}
return 0;
}
}
```

```
enter username:
Window
enter password:
12345
authentication successfull
```

11.program to find an even and odd number without using arithmetic operations

```
#include<stdio.h>
int main()
{
int n;
printf("enter a value for n= ");
scanf("%d",&n);
if(n & 1){
printf("number is odd");}
else{
printf("number is even");}
return 0;
}
```

```
enter a value for n= 24
number is even

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