1. Variable Initialization

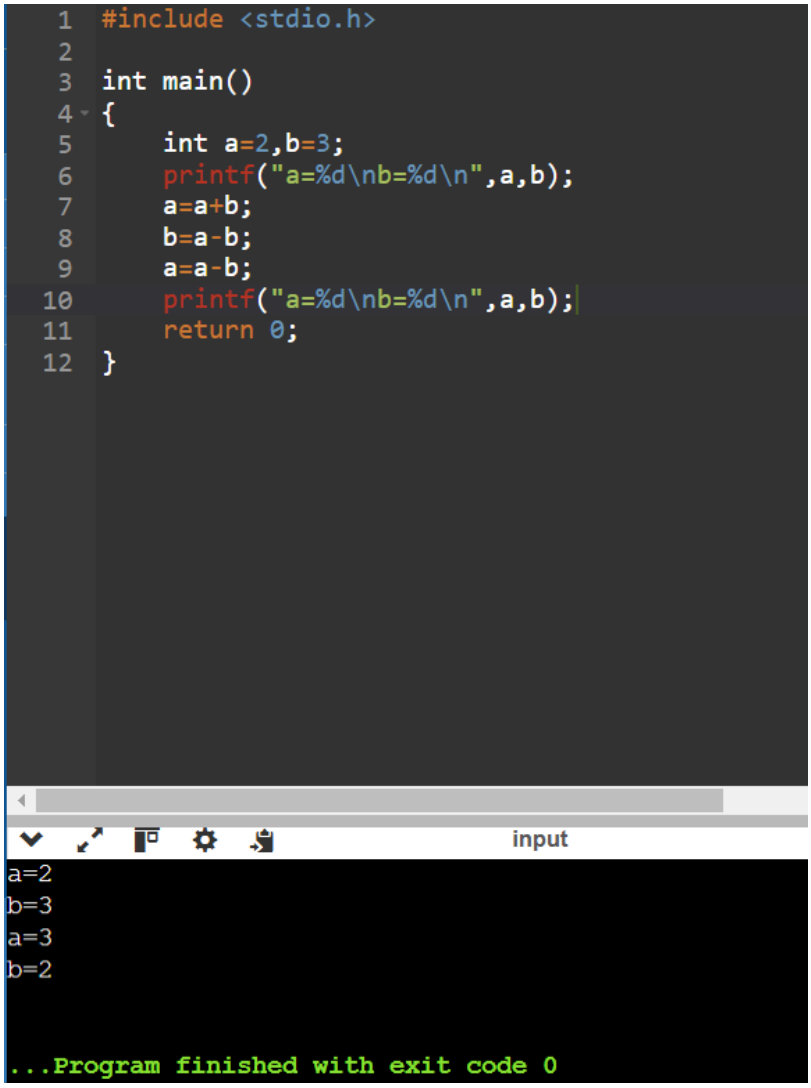
Question: Write a program that declares an integer variable, initializes it with a value of 42, and prints the value to the console.

```c
#include <stdio.h>

int main()
{
    int a=42;
    printf("%d \n",a);
    return 0;
}
```

```
input
42


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

2. Swapping Variables

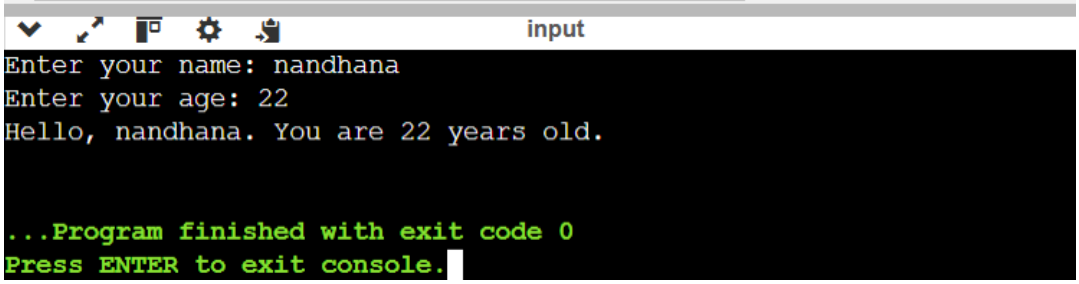
Question: Create a program that swaps the values of two integer variables without using a temporary variable. Demonstrate this by printing the values before and after the swap.



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 name[50];
    int age;
    printf("Enter your name: ");
    scanf("%s", name);
    printf("Enter your age: ");
    scanf("%d", &age);
    printf("Hello, %s. You are %d years old.\n", name, age);

    return 0;
}
```



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 a=10;
    float b;
    printf("a=%d\n",a);
    b=(float)a;
    printf("after converting,a=%f\n",b);

    return 0;
}
```

input

```
a=10
after converting,a=10.000000


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

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()
{
    int r=10;
    double area;
    area=pi*r*r;
    printf("Area of circle: %f",area);

    return 0;
}
```

input

```
Area of circle: 314.000000

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

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>

void main()
{
    int a=10;
    printf("a=%d \n",a);
    int modify()
    {
        a=5;
        return a;
    }
    modify();
    printf("a=%d",a);

}
```

input

```
a=10
a=5

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

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>

void main()
{
    int a=100;
    printf("Initial value of a=%d \n",a);
    a+= 10;
    printf("a=%d \n",a);
    a-= 20;
    printf("a=%d \n",a);
    a*= 10;
    printf("a=%d \n",a);
    a/= 20;
    printf("a=%d",a);

}
```

input

```
Initial value of a=100
a=110
a=90
a=900
a=45

...Program finished with exit code 4
```

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>

void main()
{
    int a[5]={5,10,15,20,25};
    int sum;
    for(int i=0;i<5;i++)
    {
        sum=sum+a[i];
    }
    printf("sum of all elements: %d",sum);
}
```

input

```
sum of all elements: 75

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

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

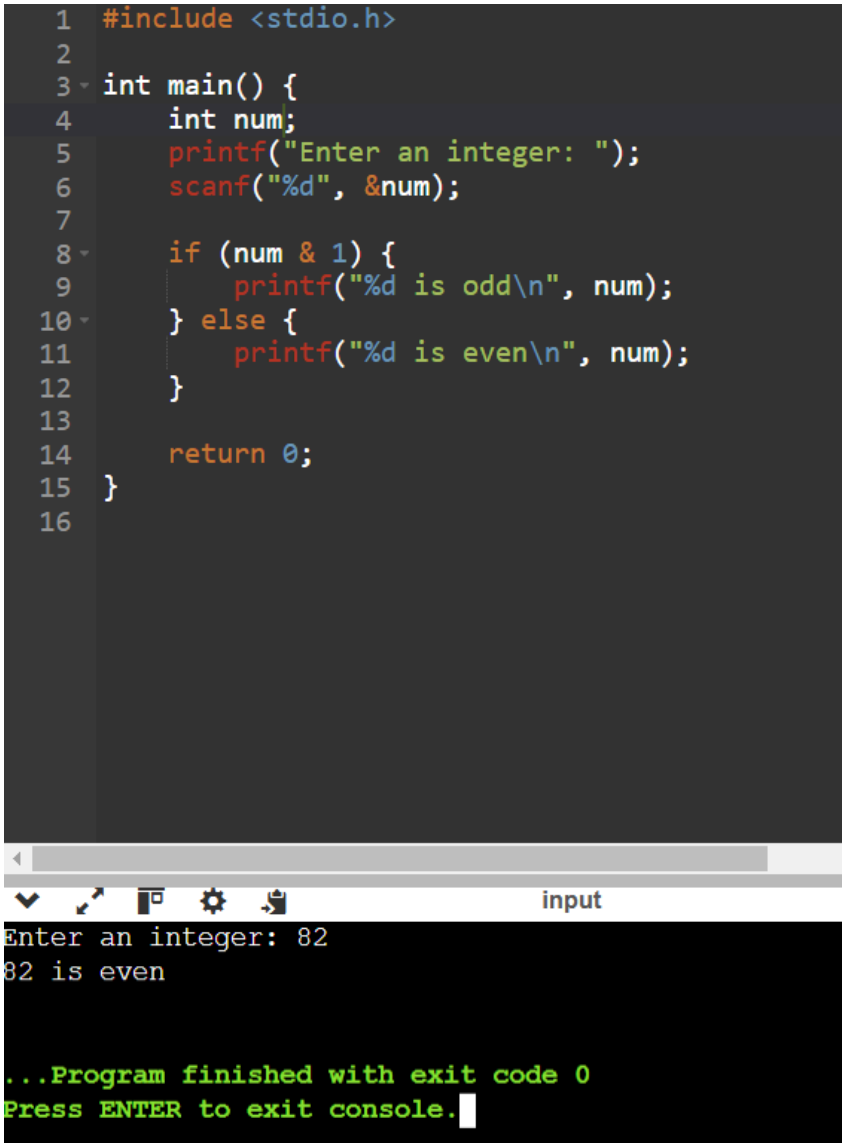
1. Define two constants for the correct username and password.
2. Prompt the user to enter their username and password.
3. Use logical operators (`&&`, `||`, `!`) to check if:
4. If both are correct, display a success message.
5. Implement additional checks:
    - If the username is empty, display a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

Answer:

```
#include <stdio.h>
#include<string.h>
#define username "user1"
#define password "1234"
int main()
{
   char u[50],p[50];
   printf("Enter username and password\n");
   scanf("%s",u);
   scanf("%s",p);

   if(u[0]=='\0'|| p[0]=='\0')
      printf("username and password cannot be empty\n");
   else if(strcmp(u,username)==0 && strcmp(p,password)==0)
      printf("username and password matches\n");
```

```
    else if(!(strcmp(u,username)==0 && strcmp(p,password)==0))
    {
        printf("Incorrect username or password\n");
    }
    return 0;
}
```

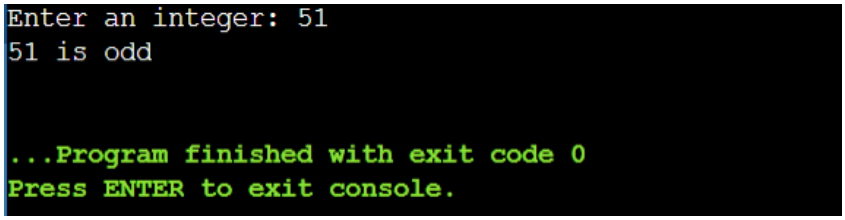
logic to check whether the number is even or odd without using any arithmetic operator



```c
#include <stdio.h>

int main() {
    int num;
    printf("Enter an integer: ");
    scanf("%d", &num);

    if (num & 1) {
        printf("%d is odd\n", num);
    } else {
        printf("%d is even\n", num);
    }

    return 0;
}
```

input

```
Enter an integer: 82
82 is even


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


Enter an integer: 51
51 is odd
...Program finished with exit code 0
Press ENTER to exit console.