Answer:

```
#include<stdio.h>
int main(){
 int a=42;
 printf("The value of a= %d \n",a);
 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.

Answer:

```
#include<stdio.h>
int main(){
 int a,b;
 printf("The value of a= \n");
 scanf("%d", &a);
 printf("The value of b= \n");
 scanf("%d", &b);
 printf("The value before swapping \n a=%d \n b= %d",a,b);
 a=a+b;
 b=a-b;
 a=a-b;
 printf("The value after swapping \n a=%d \n b= %d",a,b);
 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 name and age.

Answer:

```
#include <stdio.h>
int main() {
    int age;
    char name[50];
    printf("Enter the age: ");
    scanf("%d", &age);
    getchar();
    printf("Enter the name: ");
    scanf("%[^\n]", name);
    printf("Hello %s. Age = %d\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.

Answer:

```
#include <stdio.h>
int main() {
    int a=25;
    float b=a;
    printf("The value of a after conversion=%.2f",b);
    return 0;
}
```

----

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.

Answer:

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

----

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.

Answer:

```
#include <stdio.h>
int x = 10;
void modifyx() {
    x = 20;
    printf("Inside modifyx: x = %d\n",x);
}
int main() {
    printf("Before modification: x = %d\n", x);
    modifyx();
    printf("After modification: x = %d\n", x);
    return 0;
}
```

----

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.
Answer:

```
#include <stdio.h>
int main() {
   int num = 100;
   num += 10;
   printf("After += 10: %d\n", num);
   num -= 5;
   printf("After -= 5: %d\n", num);
   num *= 2;
   printf("After *= 2: %d\n", num);
   num /= 4;
   printf("After /= 4: %d\n", num);
   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.
Answer:

```
#include <stdio.h>
int main() {
   int arr[5] = {3, 5, 7, 2, 8};  // Initialize the array
   int sum = 0;
   for(int i = 0; i < 5; i++) {
      sum += arr[i];
   }
   printf("Sum of array elements: %d\n", sum);
   return 0;
}
```

----

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.

Answer:

```
#include <stdio.h>
#include <string.h>
#define USERNAME "admin"   // Predefined correct username
#define PASSWORD "1234"    // Predefined correct password

int main() {
    char username[20];
    char password[20];
    printf("Enter username: ");
    scanf("%s", username);
    if (strlen(username) == 0) {
        printf("Error: Username cannot be empty.\n");
        return 1;
    }
    printf("Enter password: ");
    scanf("%s", password);
    if (strlen(password) == 0) {
        printf("Error: Password cannot be empty.\n");
        return 1;
    }
    if (strcmp(username, USERNAME) == 0 && strcmp(password, PASSWORD) == 0) {
        printf("Authentication successful!\n");
    } else {
        printf("Error: Incorrect username or password.\n");
    }
    return 0;
}
```

----

11. Program to find an even number and odd number without using arithmetic operators
Answer:

```
#include<stdio.h>
int main(){
 int num;
 printf("Enter the number=");
 scanf("%d", &num);
 if((num & 1)==0){
  printf("The number is even");
 }
 else{
  printf("The number is odd");
 }
 return 0;
}
--------------------------------------------------------------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------------------
```