DAY 3 ASSIGNMENT

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;
    a=42;
    printf("a = %d",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.

```c
#include<stdio.h>
int main(){
    int a,b;
    a=10;
    b=20;
    printf("Before Swapping a = %d\n",a);
    printf("Before Swapping b = %d\n",b);
    a=a+b;
    b=a-b;
    a=a-b;
    printf("After Swapping a = %d\n",a);
    printf("After Swapping b = %d",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.

```c
#include <stdio.h>

int main() {
    char name[50];
    int age;

    printf("Please enter your name: ");
    scanf("%s",name);
    printf("Please enter your age: ");
    scanf("%d", &age);
    printf("Hello, %sYou 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;
    a=10;
    float float_a = (float)a;
    printf("a = %d",a);
    printf("Float Value of a = %.2f",float_a);
    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.

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

int main(){
    float radius,area;

    printf("Enter Radius :");
    scanf("%f",&radius);
    area=Pi*radius*radius;
    printf("Area of Circle with radius %.2f is %.2f",radius,area);
    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.

```c
int globalVar = 5;
void modifyGlobalVar() {
    globalVar = 10;
}

int main() {

    printf("Value of globalVar before modifying: %d\n", globalVar);
    modifyGlobalVar();
    printf("Value of globalVar after modifying: %d\n", globalVar);

    return 0;
}
```

7. 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 num = 10;
```

```c
    num += 10;
    printf("After += 10, num = %d\n", num);

    num -= 20;
    printf("After -= 20, num = %d\n", num);

    num *= 3;
    printf("After *= 3, num = %d\n", num);

     num /= 5;
    printf("After /= 5, num = %d\n", num);

    return 0;
}
```

8. 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 arr[5] = {3, 4, 12, 5, 10};
    int sum = 0;

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

    printf("The sum of all elements in the array is: %d\n", sum);

    return 0;
}
```

9.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:
    1. If the username is empty, display a message indicating that the username cannot be empty.
    2. If the password is empty, display a message indicating that the password cannot be empty.

3. The username matches the predefined username AND the password matches the predefined password.
4. If either the username or password is incorrect, display an appropriate error message.

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

#define USERNAME "user123"
#define PASSWORD "pass123"

int main() {
   char username[50];
   char password[50];

   printf("Enter username: ");
   scanf("%s",username);

   printf("Enter password: ");
   scanf("%s",password);

   if (strlen(username) == 0) {
      printf("Username cannot be empty\n");
   }
   else if (strlen(password) == 0) {
      printf("Password cannot be empty\n");
   }
   else if (strcmp(username, USERNAME) == 0 && strcmp(password, PASSWORD) == 0) {
      printf("Authentication successful. Welcome!!\n");
   }
   else {
      printf("Error::Incorrect username or password.\n");
   }

   return 0;
}
```

10.Operator Associativity and Precedence

```c
#include <stdio.h>

int main()
{
   int x=2;
   int y=++x + x++ + --x;
   printf("Value of y = %d",y);

}
```

Output : 10

11.Check whether a number is odd or even

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

```c
int main() {
    int num;

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

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

    return 0;
}
```

12. Type Casting

```c
#include <stdio.h>

int main(){
    float f=12.38;
    int i=f;
    printf("f = %f\n",f);
    printf("i = %d",i);
}
```