

## 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 num=42;
    printf("number=%d",num);

    return 0;
}
*****
```

OUTUT

---

## 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 num1,num2;
    num1=10;
    num2=15;
    printf("before swapping\n");
    printf("number1=%d number2=%d\n",num1,num2);
    num1=num1+num2;
    num2=num1-num2;
    num1=num1-num2;
    printf("after swapping\n");
    printf("number1=%d number2=%d\n",num1,num2);
    return 0;
}
*****
```

OUTPUT

```
before swapping
number1=10 number2=15
after swapping
number1=15 number2=10
```

---

## 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.

```
#include <stdio.h>
#include<string.h>
int main()
{
    char name[20];
    unsigned int age;
    printf("enter the name \n");
    scanf("%s",name);
    printf("enter the age");
    scanf("%u",&age);
    printf("Hello i am %s , my age is %u",name,age);
    return 0;
}
*****
```

#### OUTPUT

```
enter the name
antony
enter the age22
Hello i am antony , my age is 22
```

---

#### 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>
#include<string.h>
int main()
{
    int int_value=10;
    float float_value;
    float_value=(float)int_value;
    printf("float value is %f\n",float_value);
    printf("int value is %d",int_value);
}
*****
```

#### OUTPUT

```
float value is 10.000000
int value is 10
```

---

#### 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>
#include<math.h>
#define pi 3.14
```

```
int main(){
    unsigned int radius=20;
    float area;
    area=pi*pow(radius,2);
    printf("area of circle with radius %u is %f",radius,area);
}
*****
```

## OUTPUT

area of circle with radius 20 is 1256.000000

-----

## 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 global_var=50;
int main(){
    printf("before modification\n");
    printf("%d\n",global_var);

    global_var=60;
    printf("after modification\n");
    printf("%d",global_var);
}
*****
```

## OUTPUT

before modification

50

after modification

60

-----

## 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 num = 100;
    printf("Initial value: %d\n", num);
    num += 10;
    printf("After += 10: %d\n", num);
```

```

    num -= 20;
    printf("After -= 20: %d\n", num);
    num *= 3;
    printf("After *= 3: %d\n", num);
    num /= 2;
    printf("After /= 2: %d\n", num);
    return 0;
}
*****

```

## OUTPUT

Initial value: 100

After += 10: 110

After -= 20: 90

After \*= 3: 270

After /= 2: 135

-----

## 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 sum=0;
    int num_arr[]={10,35,25,40,10};
    for(int count=0;count<5;count++){
        sum=sum+num_arr[count];
    }
    printf("sum of elements in the array= %d",sum);

    return 0;
}
*****

```

## OUTPUT

sum of elements in the array= 120

-----

## 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 og_username "antony"
#define og_password "1234567"
int main()
{
    char password[8],username[7];
    printf("enter the username\n");
    scanf("%s",username);
    printf("enter the password\n");
    scanf("%s",password);
    if(strlen(username)==0){
        printf("username cannot be empty");
    }
    else if(strlen(password)==0){
        printf("password cannot be empty");
    }
    else if(strcmp(password,og_password)==0&& strcmp(username,og_username)==0){
        printf("username and password crct");
    }
    else{
        printf("username or password not crct");
    }
    return 0;
}
```

-----