

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.

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.

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.

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.

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.

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.

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.

1. variable initialization

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

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

Output: Value of a is: 42

2. swapping variables

```
#include<stdio.h>

int main()
{
    int a=20;
    int b=30;
    printf("Before swapping a= %d, b= %d\n",a,b);
    a=a+b;
    b=a-b;
    a=a-b;
    printf("After swapping a=%d, b=%d \n",a,b);
    return 0;
}
```

Output: Before swapping a= 20, b= 30

After swapping a=30, b=20

3.user input and output

```
#include<stdio.h>

int main()
{
    char name[20];
    int age;
    printf("Enter name:");
```

```

scanf("%s",name);
printf("Enter age:");
scanf("%d",&age);
printf("My name is %s, lam %d years old",name,age);
return 0;
}

```

Output: Enter name:anu

Enter age:20

My name is anu, lam 20 years old

4.data type conversion

```

#include<stdio.h>
int main()
{
    int a=10;
    printf("Integer value of a %d \n",a);
    float b=(float)a;
    printf("Float value of a is %f \n",b);
    return 0;
}

```

Output: Integer value of a 10

Float value of a is 10.000000

5. Constants vs variables

```

#include <stdio.h>
#define PI 3.14
int main()
{
    float radius=2,area;

```

```
    area=PI *radius*radius;

    printf("Area of circle is: %f \n",area);

    return 0;

}
```

Output: Area of circle is: 12.560000

6.scope of variables

```
#include <stdio.h>

int globalVar = 10;

void fun() {
    globalVar += 5;
}

int main() {

    printf("Global variable before modification: %d\n", globalVar);

    fun();

    printf("Global variable after modification: %d\n", globalVar);

    return 0;

}
```

Output: Global variable before modification: 10

Global variable after modification: 15

7. Using augmented assignment operator

```
#include<stdio.h>

int main()

{

    int num=100;
```

```

num+=20;

printf("+=: %d\n",num);

num-=20;

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

num*=20;

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

num/=20;

printf("/=: %d\n",num);
}

```

Output: +=: 120

-=: 100

*=: 2000

/=: 100

8. Array of variables

```

#include<stdio.h>

int main()
{
    int arr[5]={1,2,3,4,5};

    int sum=0;

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

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

    return 0;
}

```

Output: The sum of the elements:15

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.

Below is a C program that meets the requirements of your assignment for user authentication. The program defines constants for the correct username and password, prompts the user for input, and checks the credentials using logical operators. It also includes checks for empty username and password inputs.

user authentication system

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

```
#define USERNAME "anu"
```

```
#define PASSWORD "12345"
```

```
int main()
```

```
{
```

```
    char username[30];
```

```
    char password[20];
```

```
    printf("Enter your username:");
```

```
    scanf("%s",username);
```

```
    printf("Enter password:");
```

```
    scanf("%s",password);
```

```

if (username[0]=='\0'){
    printf("Username cannot be empty. \n");
}
else if(password[0]=='\0'){
    printf("Password cannot be empty \n");
}
else if(username[0]=='a' && username[1]=='n' && username[2]=='u' && password[0]=='1' &&
password[1]=='2' && password[2]=='3' && password[3]=='4' && password[4]=='5'){
    printf("Login successfull %s \n",username);
}
else{
    printf("Incorrect username and password \n");
}
return 0;

}

```

Output 1:Enter your username: anu

Enter password: 12345

Login successful anu

Output 2: Enter your username: arun

Enter password:123

Incorrect username and password

program for odd and even

Pseudocode:

Input a

If a and 1 then

 Print "Odd"

Else

 Print "Even"

End if

```
#include<stdio.h>

int main()
{
    int a;
    printf("Enter value for a=");
    scanf("%d",&a);
    if(a&1){
        printf("The number is odd");
    }else
    {
        printf("The number is even");
    }
}
```

Output: Enter value for a=40

The number is even