**Assignment Questions:**

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.

Program:

#include<stdio.h>

int main()

{

int num;

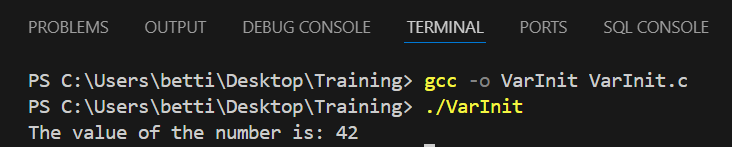
num = 42;

printf("The value of the number is: %d\n", num);

return 0;

}

Output:



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.

Program:

#include<stdio.h>

int main() {

int num1,num2;

num1 = 50;

num2 = 100;

printf("First number before swapping: %d\n", num1);

printf("Second number before swapping: %d\n", num2);

num1 = num1 + num2;

num2 = num1 - num2;

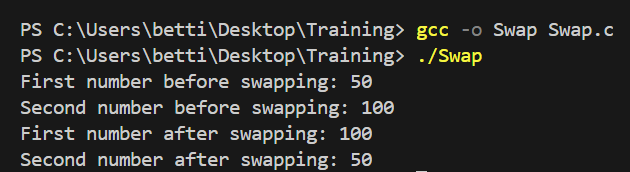
num1 = num1 - num2;

printf("First number after swapping: %d\n", num1);

printf("Second number after swapping: %d\n", num2);

return 0;

}

Output:  


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.

Program:

#include<stdio.h>

int main() {

unsigned int age;

char name[100];

printf("Enter your age: ");

scanf("%u", &age);

printf("Enter your name: ");

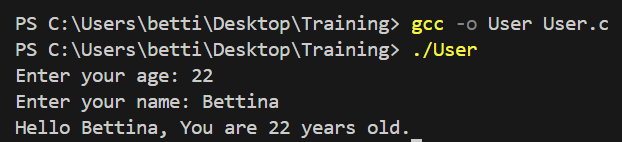
scanf("%s", name);

printf("Hello %s, You are %u years old.\n", name, age);

return 0;

}

Output:



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.

Program:

#include<stdio.h>

int main() {

int num;

num = 10;

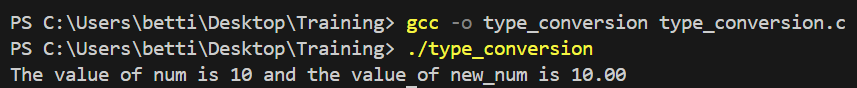
float new\_num = num;

printf("The value of num is %d and the value of new\_num is %.2f\n", num, new\_num);

return 0;

}

Output:



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.

Program:

#include <stdio.h>

#define PI 3.14

int main()

{

int r = 21;

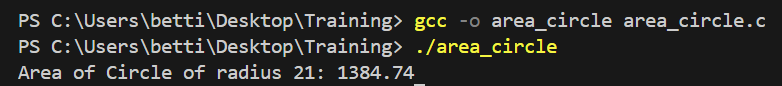
float area = PI \* r \* r;

printf("Area of Circle of radius %d: %.2f", r, area);

return 0;

}

Output:



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.

Program:

#include<stdio.h>

int a = 20;

int main() {

printf("a before modification: %d\n", a);

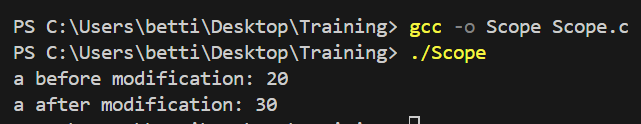
a = a + 10;

printf("a after modification: %d\n", a);

return 0;

}

Output:



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.

Program:

#include<stdio.h>

int main() {

int num = 100;

num+=50;

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

num-=100;

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

num\*=50;

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

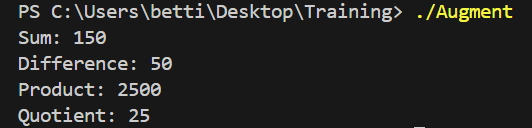
num/=100;

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

return 0;

}

Output:



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.  
Program:

#include<stdio.h>

int main() {

int sum=0;

int arr[5]={0,1,2,3,4};

printf("Array Elements: ");

for(int i=0; i<5; i++)

{

printf("%d ", arr[i]);

}

for(int i=0; i<5; i++)

{

sum+=arr[i];

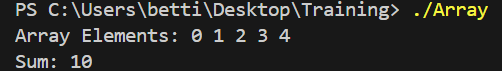
}

printf("\nSum: %d\n", sum);

return 0;

}

Output:



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: 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.  
Program:

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#define username "Bettina"

#define password "123456"

int main() {

char entered\_username[20], entered\_password[20];

printf("Enter your username: ");

scanf("%s", entered\_username);

printf("Enter your password: ");

scanf("%s", entered\_password);

if(entered\_username[0] != '\0' && entered\_password[0] != '\0')

{

if(strcmp(entered\_username, username) == 0 && strcmp(entered\_password, password) == 0) {

printf("Login successful!\n");

}

else if(strcmp(entered\_username, username)!=0) {

printf("Username is incorrect.\n");

}

else if(strcmp(entered\_password, password)!= 0) {

printf("Password is incorrect.\n");

}

}

else{

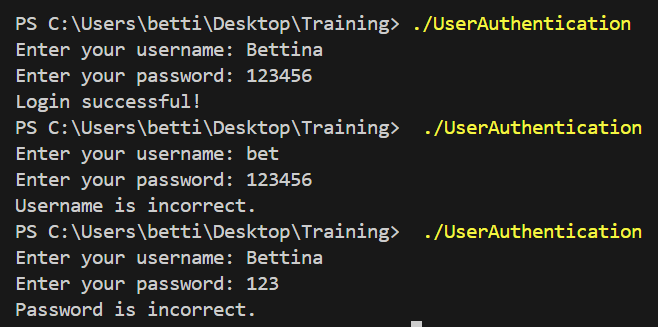
printf("Username and password cannot be empty.\n");

}

return 0;

}

Output:



**In Class Questions**

Program 1: Distance using int and char

#include<stdio.h>

int main(){

int AB = 40;

int BC = 160;

int AC = AB + BC;

printf("The total distance AC: %d\n", AC);

return 0;

}



#include<stdio.h>

int main(){

unsigned char AB = 40;

unsigned char BC = 160;

unsigned char AC = AB + BC;

printf("The total distance AC: %d\n", AC);

return 0;

}



Program 2: Finding Odd or Even using bitwise AND

#include<stdio.h>

int main()

{

int number;

printf("Enter a number: ");

scanf("%d", &number);

if(number&1){

printf("The number is odd.\n");

}

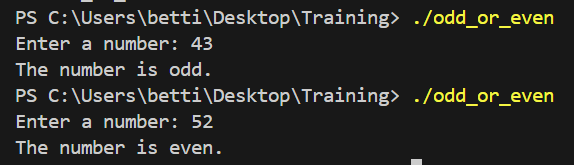
else{

printf("The number is even.\n");

}

return 0;

}



Program 3: Scope of a variable and use of static

#include<stdio.h>

void fun(void);

int flag=0;

int main()

{

fun();

fun();

fun();

return 0;

}

void fun(){

int counter = 0;

static int point = 0;

counter+=1;

flag+=1;

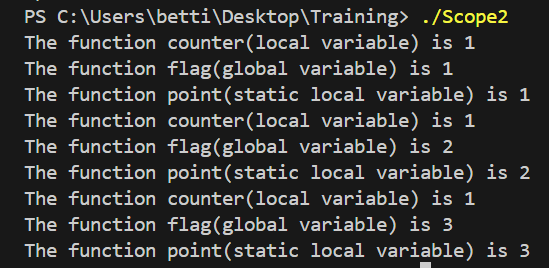
point+=1;

printf("The function counter(local variable) is %d\n", counter);

printf("The function flag(global variable) is %d\n", flag);

printf("The function point(static local variable) is %d\n", point);

}



Program 4: Type casting in C

#include<stdio.h>

int main()

{

int a=80;

int b=3;

float d = (float)a/b; //explicit type casting in a and implicit type casting in b

printf("The result of division is: %.2f\n", d);

return 0;

}



Program 5: Operator Precedence and Associativity

#include<stdio.h>

int main()

{

int x = 2;

int y = ++x + x++ + --x;

printf("The result is: %d\n", y);

}



Program 6: Operators in C

#include<stdio.h>

int main()

{

unsigned int a = 40, b = 20;

printf("sum of %d and %d is: %d\n", a, b,a+b);

printf("difference of %d and %d is: %d\n",a,b,a-b);

printf("product of %d and %d is: %d\n",a,b,a\*b);

printf("division of %d by %d is: %d\n",a,b,a/b);

printf("modulo %d by %d is: %d\n",a,b,a%b);

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

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

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

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

printf("a&b : %u\n",a&b);

printf("a|b : %u\n",a|b);

printf("a^b : %u\n",a^b);

printf("a&&b : %u\n",a&&b);

printf("a||b : %u\n",a||b);

return 0;

}

