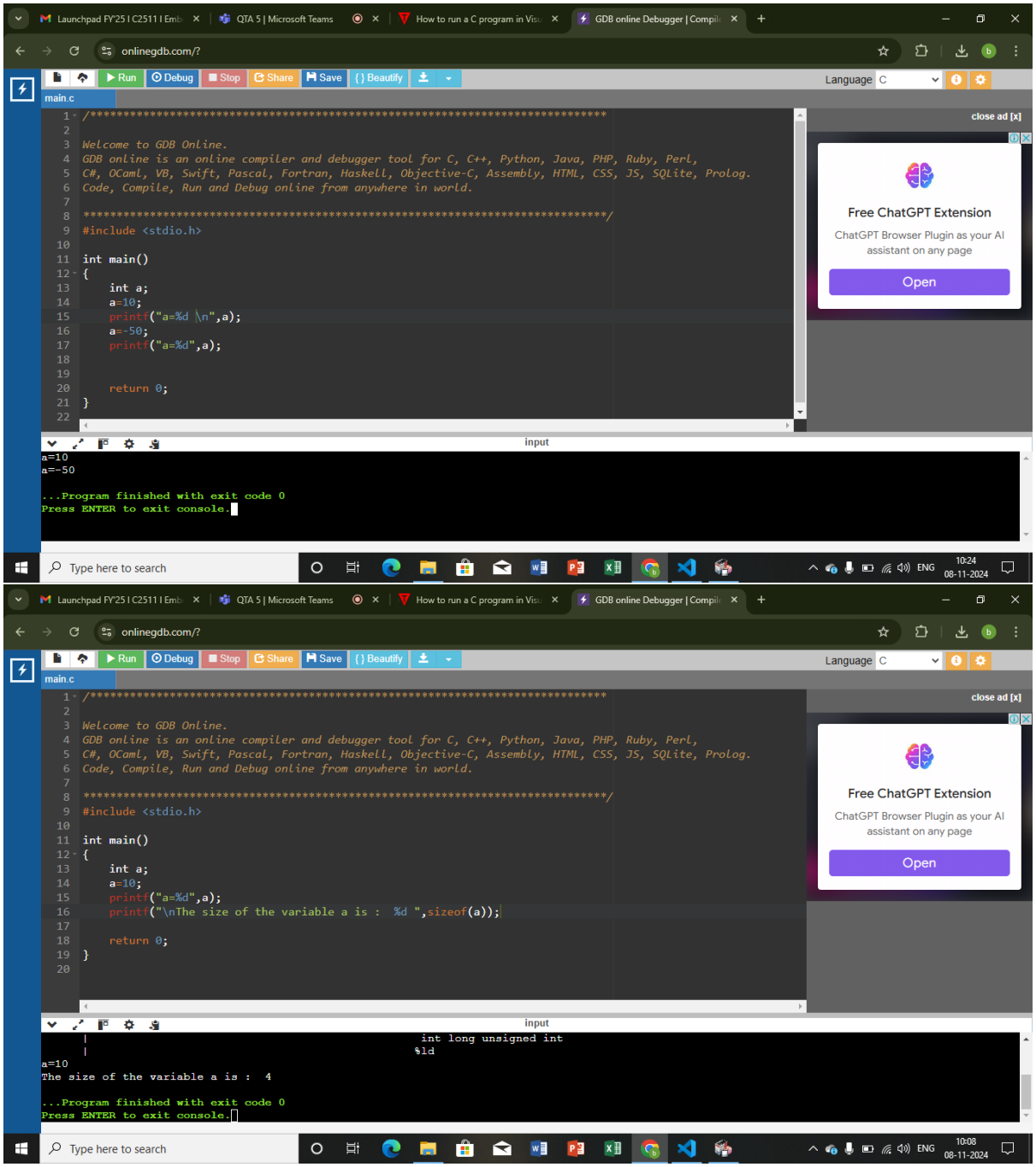
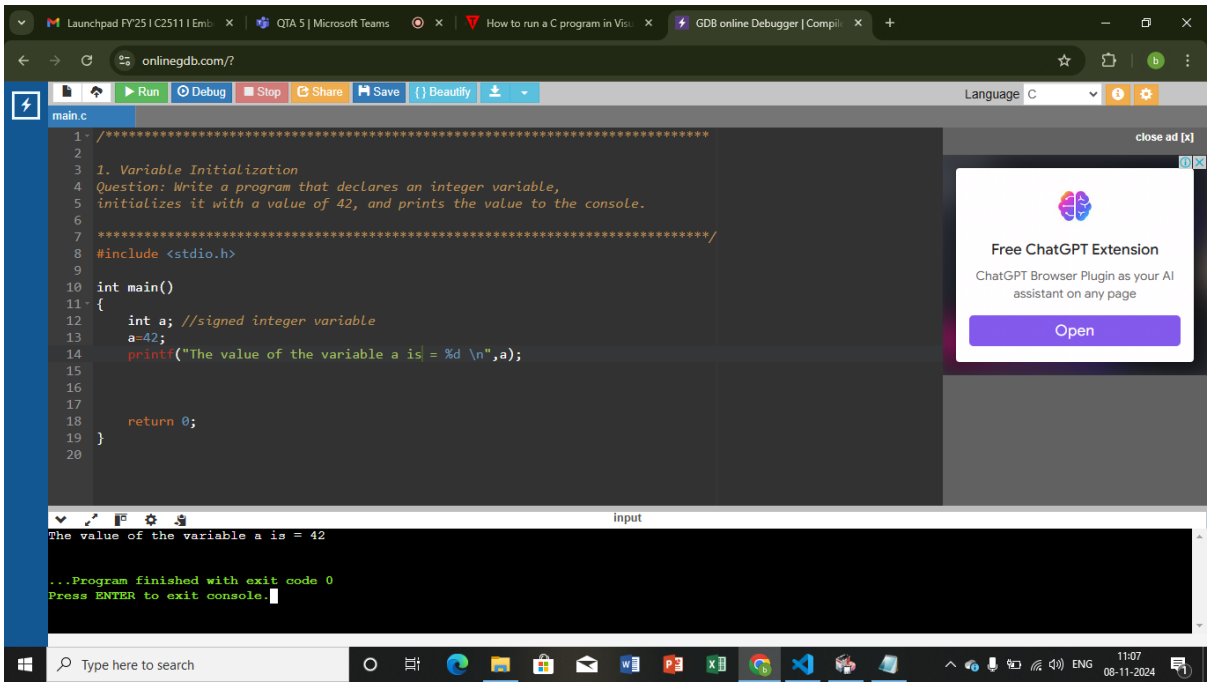


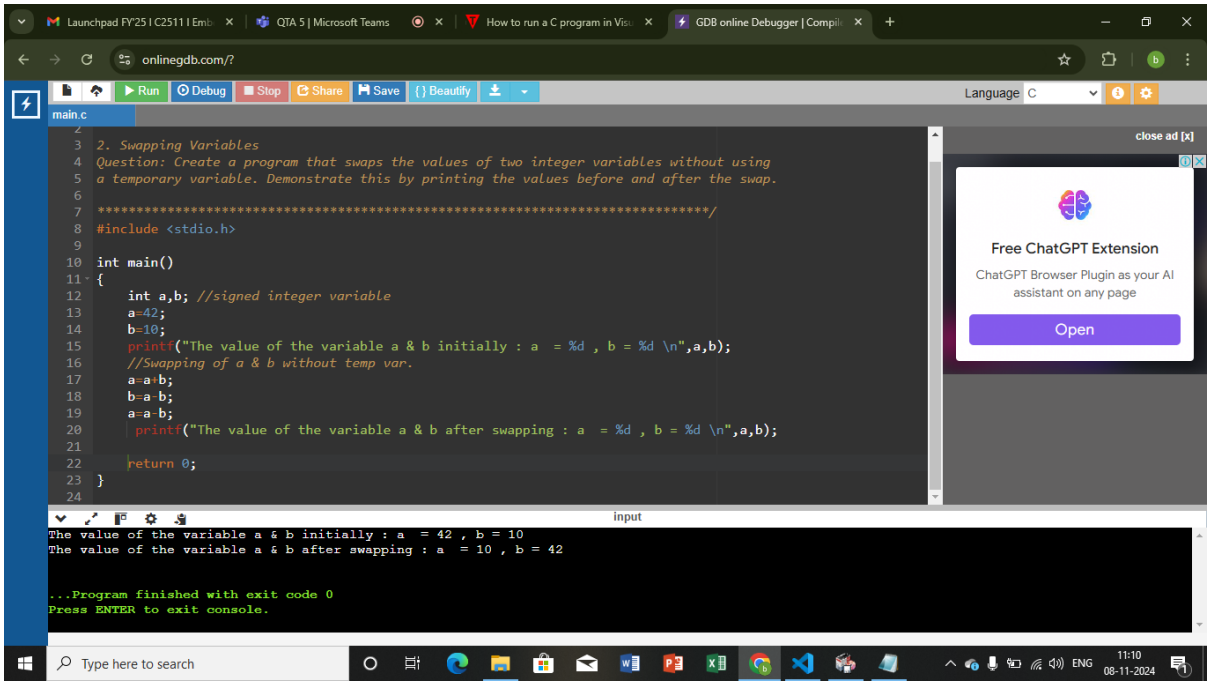
1. Variable Initialization

Question: Write a program that declares an integer variable, initializes it with a value of 42, and prints the value to the console.



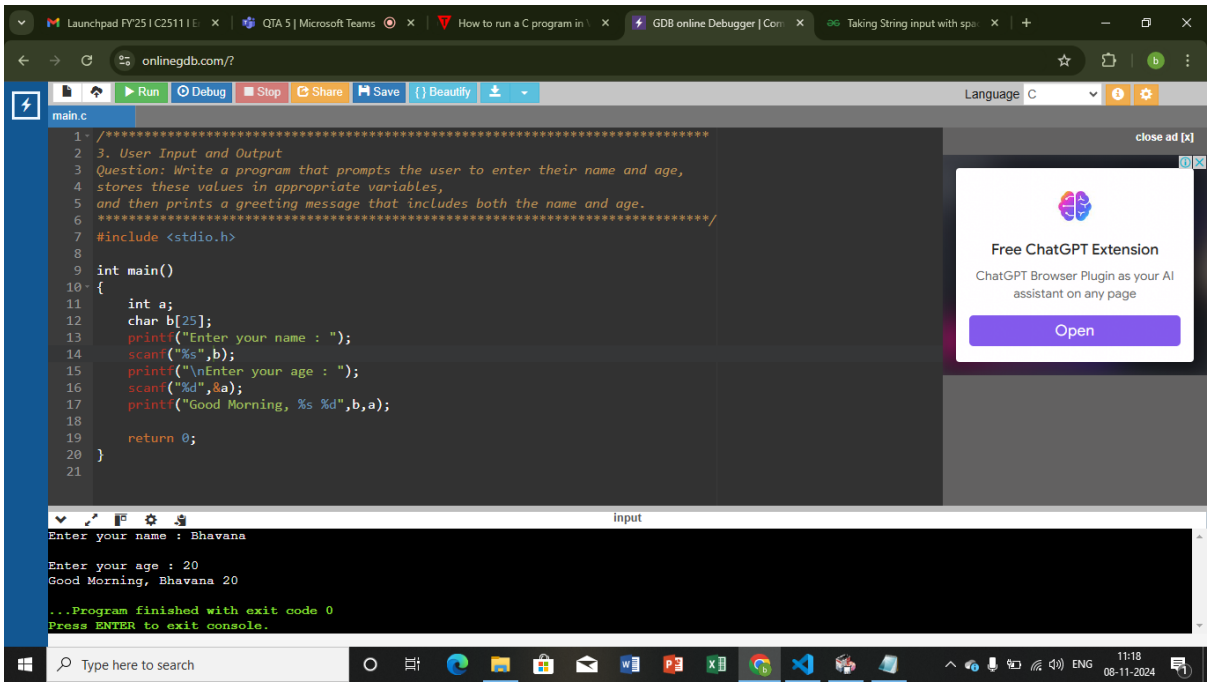
2. Swapping Variables

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.



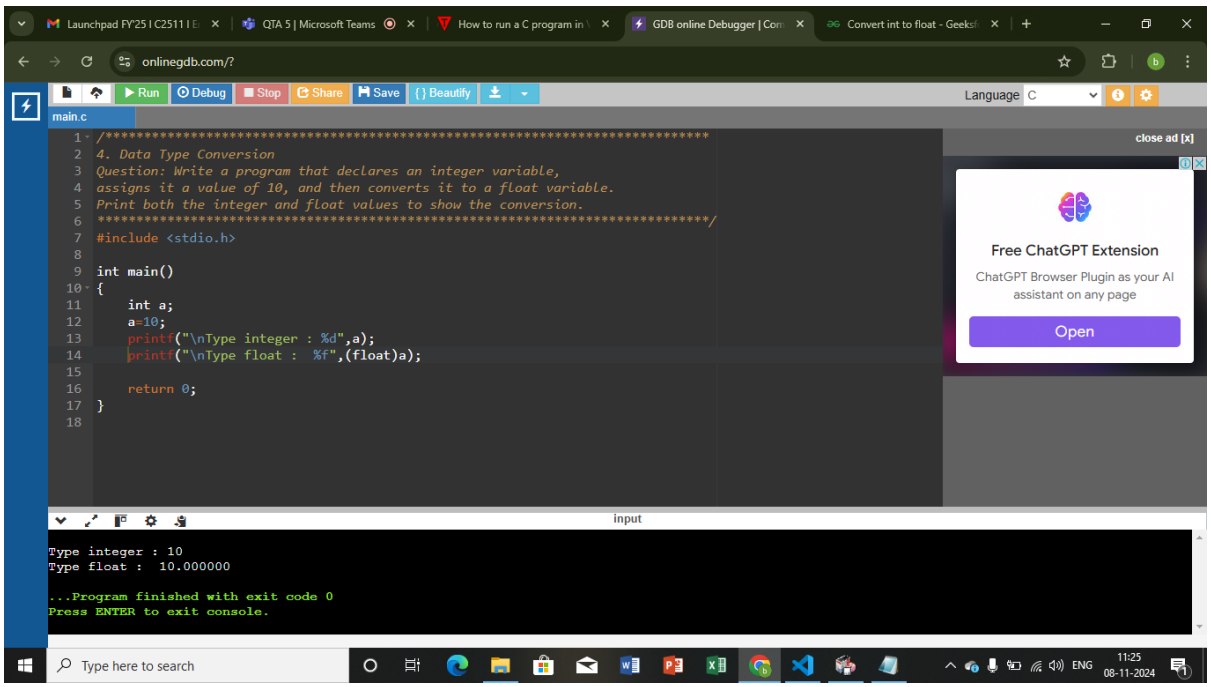
3. User Input and Output

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.



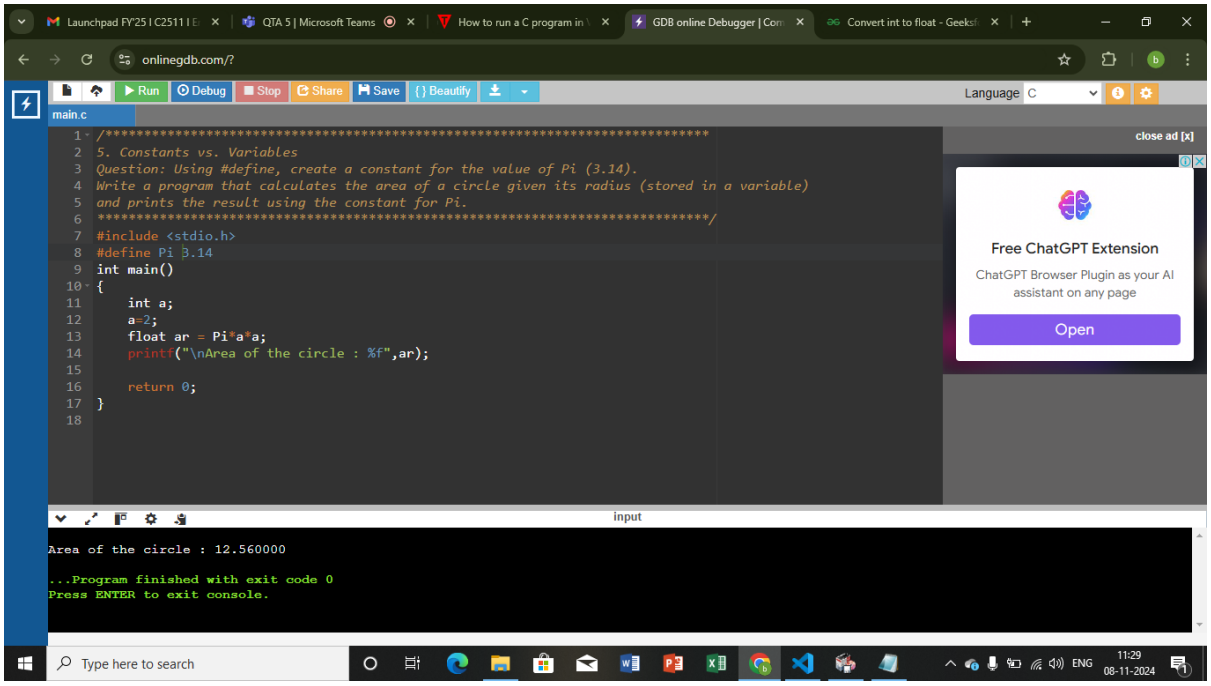
4. Data Type Conversion

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.



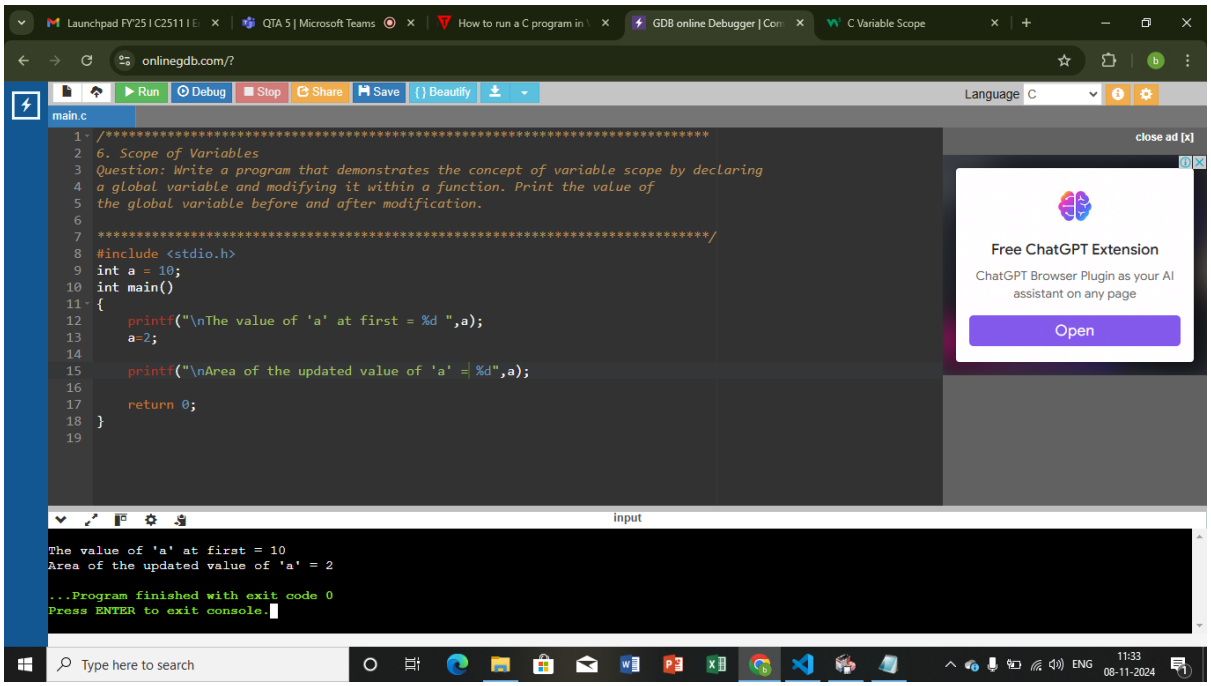
5. Constants vs. Variables

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.



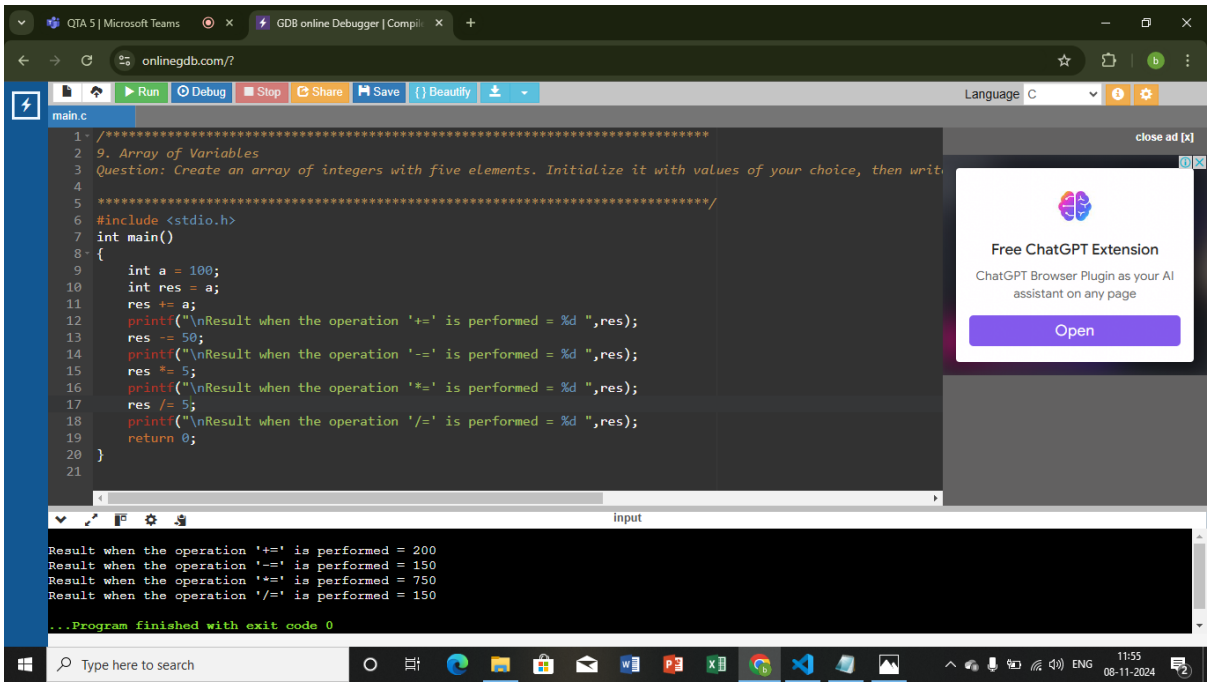
6. Scope of Variables

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.



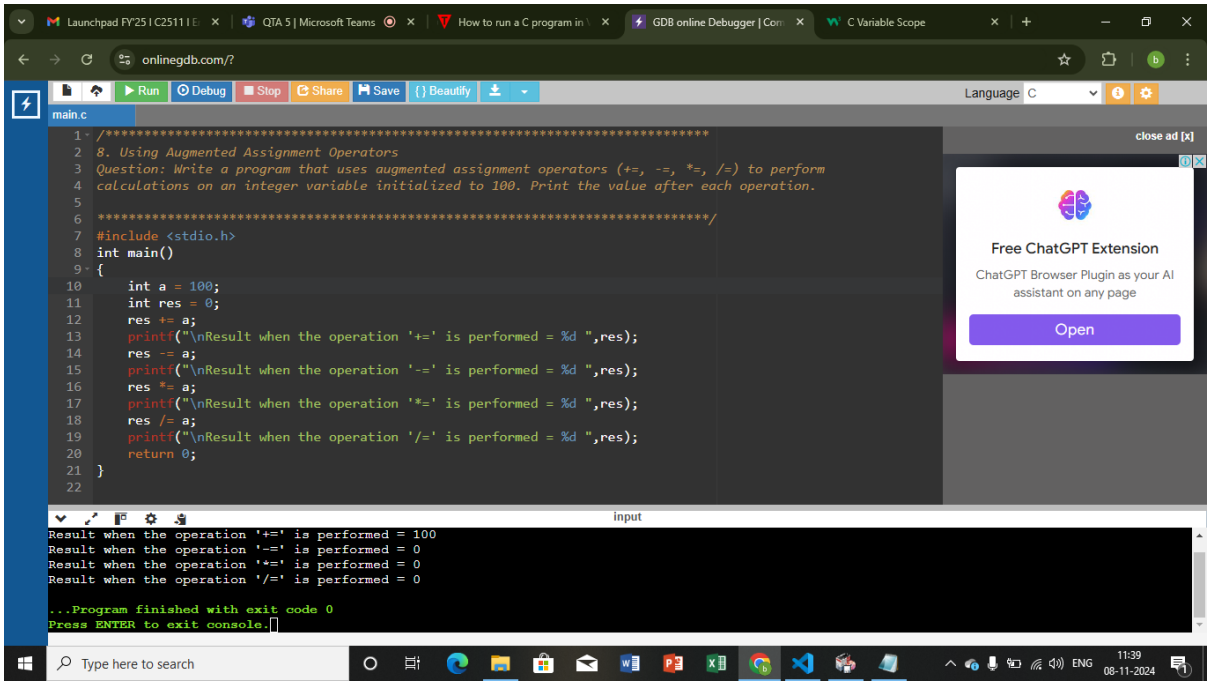
7. Using Augmented Assignment Operators

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.



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 9.

```
/** 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.
**/
#include <stdio.h>
#include <string.h>
#define nam "Maya"
#define passwrd "password"
int main()
{

    char name[25];
    char pass[15];

    printf("Enter your name : ");
    scanf("%s",name);
    if(strlen(name)==0)
    {
        printf("\nWarning : The username cannot be empty.");

    }
    printf("\nEnter your password : ");
    scanf("%s",pass);

    if(strlen(pass)==0)
    {
        printf("\nWarning : The password cannot be empty.");
    }
```

```
    if((strcmp(name,nam)==0) && (strcmp(pass,passwrd)==0))
    {
        printf("\nAuthentification completed successfully!!!");
    }
    else{
        printf("Invalid username or password.");
    }


}
```

OUTPUT :

1) Enter your name : MAYA

Enter your password : password

Invalid username or password.

2) Enter your name : Maya

Enter your password : password

Authentification completed successfully!!!

PROGRAM 10 :

```
/*Logic to check whether the number is even or odd .

Condition : Don't use any arithmetic operator.
PSEUDOCODE:
1.  N ← Read input
2.   If N & 1 == 0 then
     a.  Print "Number is even"
3.   Else
     a.  Print "Number is odd"
*/
#include <stdio.h>
int main()
{
    int a ;
    printf("Enter the number : ");
    scanf("%d",&a);
    if((a&1)==0)
    {
        printf("\nThe number is even");

    }
    else{
        printf("\nThe number is odd");
```

```
    }
}
```

OUTPUT : 1) Enter the number : 26

The number is even

2) Enter the number : 23

The number is odd

PROGRAM 11 :

```c
#include <stdio.h>

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

    return 0;
}
```

OUTPUT : Value of y is 10.

PROGRAM 12 :

```c
#include <stdio.h>
void fun(void);
int count = 0;
int main()
{
    fun();
    fun();
    fun();
    return 0;
}
void fun()
{

    count=count+1;
    printf("\nThe num = %d",count);
}
```

OUTPUT :

The num = 1

The num = 2

The num = 3

PROGRAM 13 :

```c
#include <stdio.h>
int main()
{
    int a= 40,b = 24;
    printf("Sum = %d",(a+b));
    printf("\nDifference = %d",(a-b));
    printf("\nProduct = %d",(a*b));
    printf("\nDivision = %d",(a/b));
    printf("\nincrement of a = %d",a++);
    printf("\nincrement of b = %d",b++);
    printf("\ndecrement of a = %d",a--);
    printf("\ndecrement of b = %d",b--);
    printf("\ndecrement of a = %d",--a);
    printf("\ndecrement of b = %d",--b);
    int c = 15;
    a=5;
    b=10;
    int result = a + b * c /b -a;
    printf("\nResult = %d",result);
    return 0;
}
```

OUTPUT :

Sum = 64

Difference = 16

Product = 960

Division = 1

increment of a = 40

increment of b = 24

decrement of a = 41

decrement of b = 25

decrement of a = 39

decrement of b = 23

Result = 15

PROGRAM 14 :

```c
#include <stdio.h>
int main()
{
    char A = 10;
```

```
    printf("The value of A is : %d",A);

    return 0;
}
```

OUTPUT :

The value of A is : 10

PROGRAM 15 :

```
#include<stdio.h>
int sum(int,int);
int main()
{
    int a = 20;
    int b =30;
    int y = sum(a,b);
    printf("Sum = %d",y);
    return 0;
}
int sum(int c,int d)
{
    int sum = 0;
    sum = c + d;
    return sum;
}
```

OUTPUT :

Sum = 50