

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 a;

    a=50;

    printf("the value of a is %d",a);
}
```



```
● the value of a is 50
○ PS D:\learning c\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.

```
#include<stdio.h>

int main()
{
    int a=10;

    int b=30;


    printf("The value of a before swap is %d\n", a);
    printf("The value of b before swap is %d\n", b);


    a = a * b;
    b = a / b;
    a = a / b;


    printf("After swap, a is %d\n", a);
    printf("After swap, b is %d\n", b);
}
```

```
    return 0;
}
```

```
The value of a before swap is 10
The value of b before swap is 30
After swap, a is 30
After swap, b is 10
PS D:\learning c\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.

```
#include<stdio.h>

int main()
{
    int age;
    char name[10];
    printf("enter your name");
    scanf("%s",&name);
    printf("enter your age");
    scanf("%d",&age);
    printf("%s is %d years old",name,age);

}
```

```
PS D:\learning c\output> & .\'assignment3.exe'
enter your name rinta
enter your age 22
rinta is 22 years old
PS D:\learning c\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.

```
#include<stdio.h>

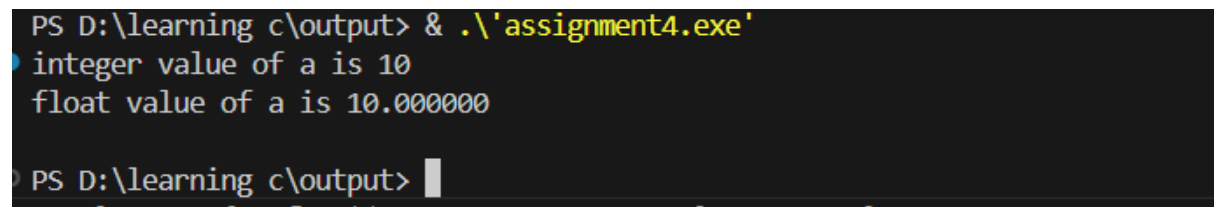
int main()
{
    int a=10;

    printf("integer value of a is %d \n",a);

    float b=(float) a;

    printf("float value of a is %f \n ",b);

    return 0;
}
```



```
PS D:\learning c\output> & .\'assignment4.exe'
integer value of a is 10
float value of a is 10.000000
PS D:\learning c\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. #include<stdio.h>

```
#define pi 3.14

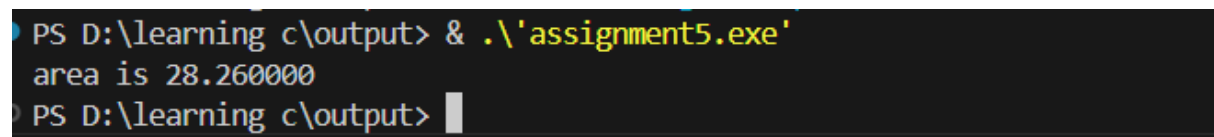
int main()
{

    float radius=3;

    float result=pi*radius*radius;

    printf("area is %f",result);

}
```



```
PS D:\learning c\output> & .\'assignment5.exe'
area is 28.260000
PS D:\learning c\output> 
```

6. Scope of Variables

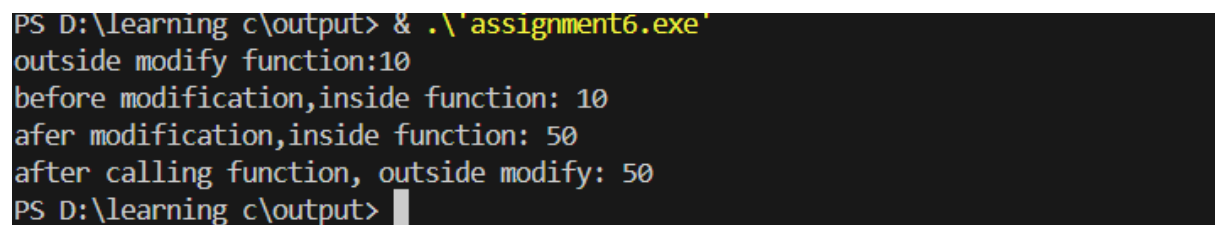
Question: Write a program that demonstrates the concept of variable scope by declaring a global variable and modifying it within a function. Print the value of the global variable before and after modification.

```
#include<stdio.h>

int globalvar=10;

int modify()
{
    printf("before modification,inside function: %d \n",globalvar);
    globalvar=50;
    printf("after modification,inside function: %d \n",globalvar);
    return globalvar;
}

int main()
{
    printf("outside modify function:%d \n",globalvar);
    modify();
    printf("after calling function, outside modify: %d \n",globalvar);
    return 0;
}
```



```
PS D:\learning c\output> & .\'assignment6.exe'
outside modify function:10
before modification,inside function: 10
after modification,inside function: 50
after calling function, outside modify: 50
PS D:\learning c\output> █
```

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 a=100;

    printf("value of a before operation : %d \n ",a);

    a+=20;

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

    a-=30;

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

    a*=2;

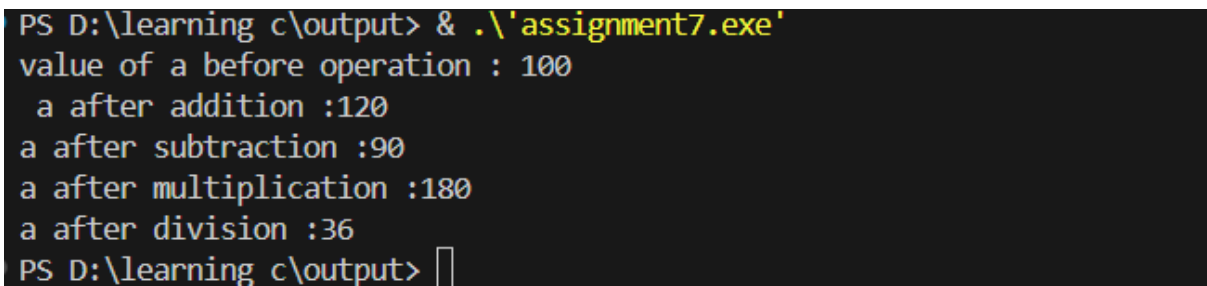
    printf("a after multiplication :%d \n",a);

    a/=5;

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

}

```



```

PS D:\learning c\output> & .\'assignment7.exe\'
value of a before operation : 100
a after addition :120
a after subtraction :90
a after multiplication :180
a after division :36
PS D:\learning c\output> 

```

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 arr[5] = {2, 4, 6, 8, 10};

    int sum = 0;

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

        sum += arr[i];
    }
}

```

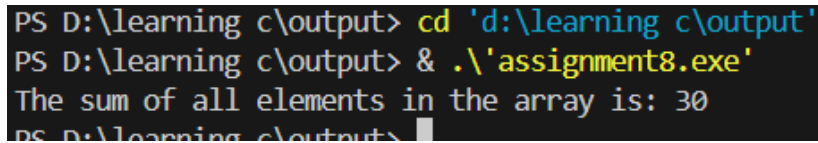
```

}

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

return 0;
}

```



```

PS D:\learning c\output> cd 'd:\learning c\output'
PS D:\learning c\output> & .\'assignment8.exe'
The sum of all elements in the array is: 30
PS D:\learning c\output>

```

10. 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 correctuser "rinta"

#define correctpass "1234rin"

int main()
{
    char username[10],password[10];

    printf("enter a username");

    scanf("%s",username);

    printf("enter password");

    scanf("%s",password);

    if(username[0]!='\0' || password[0]!='\0')
    {

```

```
    printf("enter a value");
}
if (strcmp(username,correctuser)!=0 || strcmp(password,correctpass)!=0) {
    printf("failed authentication");
}
else if(strcmp(username,correctuser)==0 && strcmp(password,correctpass)==0)
{
    printf("authentication successful");
}

}
```

```

PS D:\learning c\output> & .\'assignment9.exe'
enter a username rin
enter password 1234 rin
authentication successful
PS D:\learning c\output> cd 'd:\learning c\output'
PS D:\learning c\output> & .\'assignment9.exe'
enter a username rin
enter password 1234 rin
failed authentication
PS D:\learning c\output> cd 'd:\learning c\output'
PS D:\learning c\output> & .\'assignment9.exe'
enter a username rin
enter password 1234 rem
failed authentication
PS D:\learning c\output>

```

CLASS WORK

1. #include<stdio.h>

int main()

{

int a=10;

printf(" value of a is %d \n",a);

a=-50;

printf(" value of a is %d \n",a);

printf("size of a is %d \n ",sizeof(a));

unsigned int b=-56;

printf("b is %u \n",b);

return 0;

}

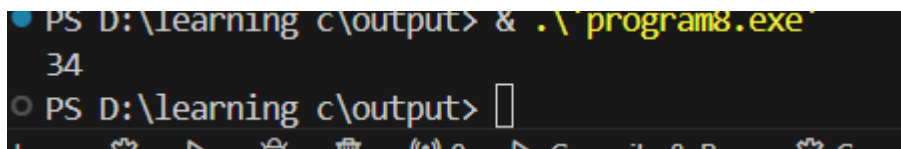
```

PS D:\learning c\output> & .\'program1.exe'
value of a is 10
value of a is -50
size of a is 4
b is 4294967240
PS D:\learning c\output>

```


2. #include<stdio.h>

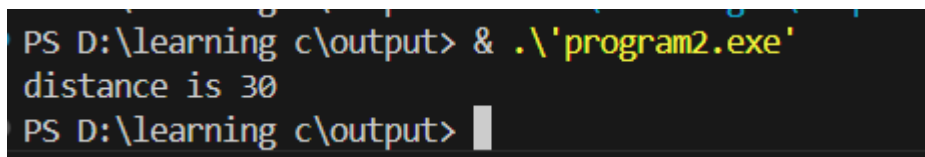
```
int main()
{
    unsigned char A = 290;
    printf("%u",A);
    return 0;
}
```



```
PS D:\learning c\output> & .\program8.exe
34
PS D:\learning c\output>
```

3. #include<stdio.h>

```
int main()
{
    unsigned char ab=20;
    unsigned char bc=10;
    unsigned char ac=ab+bc;
    printf("distance is %u",ac);
}
```



```
PS D:\learning c\output> & .\program2.exe
distance is 30
PS D:\learning c\output>
```

4. #include<stdio.h>

```
int main(){
    char A = 'h';

    printf("%d\n",A);
    printf("%c\n",A);

}
```

```
PS D:\learning c\output> cd 'd:\learning c\output'
PS D:\learning c\output> & .\program9.exe
104
h
PS D:\learning c\output> 
```

5. #include<stdio.h>

```
int main(){
```

```
char b = 20;
```

```
printf("%d\n",b);
```

```
printf("%c\n",b);
```

```
}
```

```
PS D:\learning c\output> & .\program10.exe
20
9
```

6. #include<stdio.h>

```
int main()
```

```
{
```

```
int a=5,b=10,c=15;
```

```
printf("addition %d\n",a+b);
```

```
printf("subtraction %d\n",a-b);
```

```
printf("multiplication %d\n",a*b);
```

```
printf("Division %d\n",a/b);
```

```
int result=a+b*c/b-a;
```

```
printf("the result is %d",result);
```

```
return 0;
```

```
}
```

```
PS D:\learning c\output> & .\'program3.exe'
addition 15
subtraction -5
multiplication 50
Division 0
the result is 15
```

7. #include<stdio.h>

```
int main()
{ int x=2;
int y=++x + x++ + --x;
printf("y is %d",y);
}
```

```
PS D:\learning c\output> cd 'd:\learning c\output'
PS D:\learning c\output> & .\'program4.exe'
y is 10
PS D:\learning c\output>
```

8. #include<stdio.h>

```
int main()
{
    int a=40 , b=30;
    printf("bitwise and result %d",a&b);
    printf("logical and result %d",a&&b);
}
```

```
PS D:\learning c\output> cd 'd:\learning c\output'
PS D:\learning c\output> & .\'program5.exe'
bitwise and result 8logical and result 1
PS D:\learning c\output>
```

9. //checking odd or even using bitwise operator

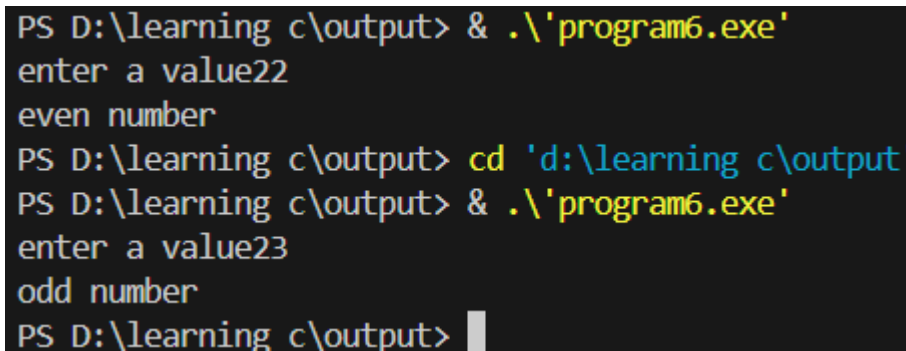
#include<stdio.h>

```
int main()
{
    int a;
```

```

printf("enter a value");
scanf("%d",&a);
if(a&1)
{
    printf("odd number");
}
else{
    printf("even number");
}
}

```



```

PS D:\learning c\output> & .\'program6.exe'
enter a value22
even number
PS D:\learning c\output> cd 'd:\learning c\output'
PS D:\learning c\output> & .\'program6.exe'
enter a value23
odd number
PS D:\learning c\output>

```

10. #include<stdio.h>

```

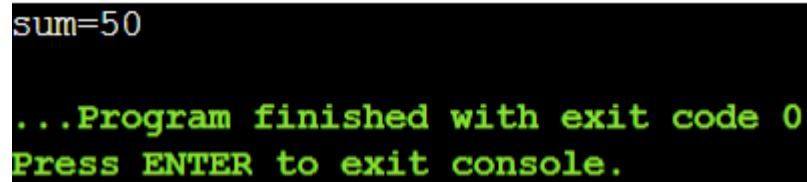
int sum(int,int);

int main()
{
    int a=20;
    int b=30;
    int y=sum(20,30);
    printf("sum=%d",y);
    return 0;
}

int sum(int c,int d)
{
    static int sum=0;
    sum=c+d;
}

```

```
    return sum;
}
```



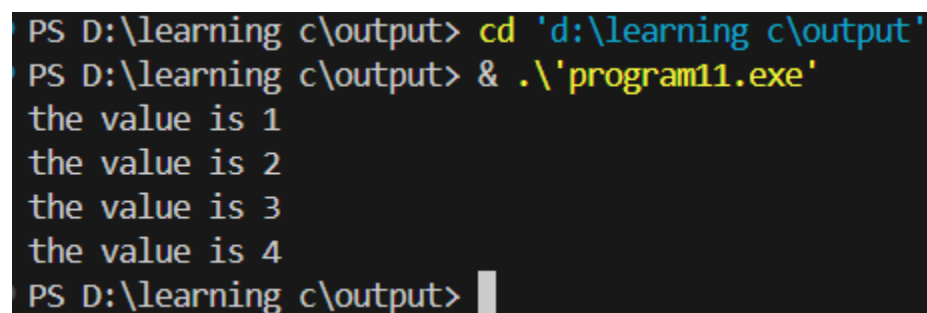
```
sum=50
...Program finished with exit code 0
Press ENTER to exit console.
```

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

void fun(void);

int main(){
    fun();
    fun();
    fun();
    fun();
}

void fun(){
    static int counter =0;
    counter = counter + 1;
    printf("the value is %d\n",counter);
}
```

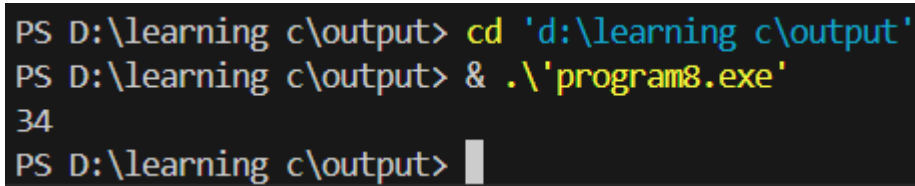


```
PS D:\learning c\output> cd 'd:\learning c\output'
PS D:\learning c\output> & .\'program11.exe'
the value is 1
the value is 2
the value is 3
the value is 4
PS D:\learning c\output>
```

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

int main()
{
    unsigned char A = 290;
    printf("%u",A);
}
```

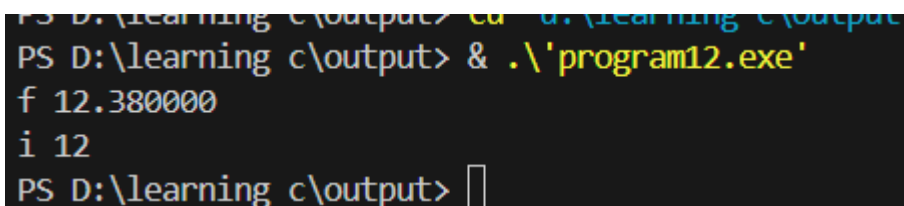
```
return 0;
}
```



A terminal window with a black background and yellow text. The prompt is 'PS D:\learning c\output>'. The first command is 'cd 'd:\learning c\output'', which is executed. The second command is '& .\'program8.exe'', which is also executed. The output of the second command is '34'. The prompt returns to 'PS D:\learning c\output>'.

13. #include<stdio.h>

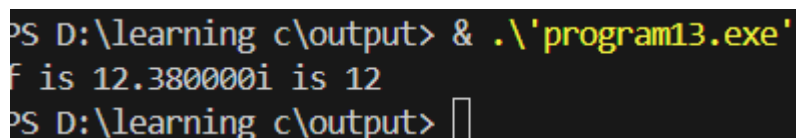
```
int main(){
float f=12.38;
int i = f;
printf("f %f\n",f);
printf("i %d",i);
}
```



A terminal window with a black background and yellow text. The prompt is 'PS D:\learning c\output>'. The command '& .\'program12.exe'' is executed. The output is 'f 12.380000' followed by 'i 12' on the next line. The prompt returns to 'PS D:\learning c\output>'.

14. #include<stdio.h>

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



A terminal window with a black background and yellow text. The prompt is 'PS D:\learning c\output>'. The command '& .\'program13.exe'' is executed. The output is 'f is 12.380000i is 12' on the same line. The prompt returns to 'PS D:\learning c\output>'.

15. #include<stdio.h>

int main()

{

int a=80;

int b=3;

float d=a/b;

printf("d = %f\n",d);

float e=(float)a/b;

printf("e =%f",e);

}

```
PS D:\learning c\output> & .\'program14.exe'
```

```
d = 26.000000
```

```
e =26.666666
```

```
PS D:\learning c\output> █
```