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

```c
#include <stdio.h>
int main()
{
  int a = 42;
  printf("a=%d", a);
}
```

Output

a=42

2)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.

```c
#include <stdio.h>
int main()
{
  int a = 5, b = 6;
  printf("the value of a=%d and b=%d before swap \n", a, b);
  a = a + b;
  b = a - b;
  a = a - b;
  printf("the value of a=%d and b=%d after swap ", a, b);
}
```

Output

```
the value of a=5 and b=6 before swap the value of a=6 and b=5 after
swap
```

3)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.

```c
#include <stdio.h>
int main()
{
  char name[50];
  int age;
  printf("enter your name:");
  scanf("%s", name);
```

```
  printf("enter your age:");
  scanf("%d", &age);
  printf("hello %s and your age is %d", name, age);
  return 0;
}
```

Output

Enter your name:navya
Enter your age:22
Hello navya and your age is 22

4)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("a=%d\n", a);
  float b = (float)a;
  printf("b=%f", b);
  return 0;
}
```

Output
a=10
b=10.000

5)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()
{
    int radius=5;
    int area=pi*radius*radius;
    printf("area=%d",area);
}
```

Output

area=78

6) 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 main()
{
   int a=50;
   printf("value of a outside function =%d\n",a);
   void global()
   {
     int a=60;
     printf("value of a inside function =%d\n",a);
   }
   global();
}
```

output

value of a outside function :50
value of a inside function :60

7)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;
   char operator;
   printf("enter operator");
   scanf(" %c",&operator);
   switch(operator)
   {
      case '+':
            a+=10;
            printf("value of a after addition is %d",a);
            break;
      case '-':
            a-=10;
            printf("value of a after substraction is %d",a);
            break;
      case '*':
             a*=10;
             printf("value of a after multiplication is %d",a);
            break;
```

```
        case '/':
               a/=10;
               printf("value of a after divison is %d",a);
              break;
        default:
              printf("invalid");
              break;
    }
}
```

8)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 a[10];
    int sum=0;
    printf("enter array elements");
    for(int i=0;i<5;i++)
    {
        scanf("%d",&a[i]);
    }
    for(int i=0;i<5;i++)
    {
        sum=sum+a[i];

    }
     printf("sum is %d",sum);
}
```

Output

```
enter array elements
4
5
6
2
3
Sum is 20
```

9)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>

int main() {
    char username[20];
    char password[20];
    printf("Enter your username: ");
    fgets(username, sizeof(username), stdin);
    printf("Enter your password: ");
    fgets(password, sizeof(password), stdin);
    if (username[0] == '\0') {
        printf("Please enter your username.\n");
    } else if (password[0] == '\0') {
        printf("Please enter your password.\n");
    } else if (strcmp(username, "navya") != 0 || strcmp(password, "1234") != 0) {
        printf("Your username or password is incorrect.\n");
    } else {

        printf("Welcome, %s!\n", username);
    }

    return 0;
}
```

Output

```
Enter your username:navya
Enter your password:1234
Welcome navya
```

10)odd or even

```
#include <stdio.h>

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

    return 0;
}
```

Output
Enter value of a:40
Even