Assignment 1: Constant Variable Declaration

Objective: Learn to declare and initialize constant variables.Write a program that declares a constant integer variable for the value of Pi (3.14) and prints it. Ensure that any attempt to modify this variable results in a compile-time error.

```
#include<stdio.h>
int main(){
    const float Pi=3.14;
    printf("Value of pi=%.2f",Pi);


    // Pi=20;
    // printf("pi=%.2f",Pi);
}
```

Assignment 2: Using const with Pointers

Objective: Understand how to use const with pointers to prevent modification of pointed values.Create a program that uses a pointer to a constant integer. Attempt to modify the value through the pointer and observe the compiler's response.

```
#include<stdio.h>
int main(){
    int const *data=10;


    //when we use const *data --->  content can be modified. but we cannot modify the address


    printf("Value of data before modification  is %d\n",data);
    data=20;
    printf("Value of data after modification  is %d",data);



}
```

Assignment 3: Constant Pointer

Objective: Learn about constant pointers and their usage.

Write a program that declares a constant pointer to an integer and demonstrates that you cannot change the address stored in the pointer.

```
#include <stdio.h>

int main() {

    int a = 10;

    int b = 20;

    int *const ptr = &a;

    // using *const ptr we can modify the address but cannot modify the data inside the ptr


    printf("Address stored in ptr: %p\n", (void*)ptr);

    // ptr = &b;

    // printf("Address stored in ptr after modification: %p\n", (void*)ptr);

    return 0;

}
```

Assignment 4: Constant Pointer to Constant Value

Objective: Combine both constant pointers and constant values.

Create a program that declares a constant pointer to a constant integer. Demonstrate that neither the pointer nor the value it points to can be changed.

```
#include <stdio.h>

int main() {

    int a = 10;

    int b = 20;

    int const *const ptr = &a;

    //using const *const ptr we cannot modify the adress as well as the content of the ptr


    printf("Value of pointer: %d\n", *ptr);
    printf("Address of pointer: %p\n", (void*)ptr);

    // *ptr=20;

    // printf("Value of pointer after modification: %d\n", *ptr);

    // ptr = &b;

    // printf("Address of pointer after modification: %p\n", (void*)ptr);

    return 0;

}
```

Assignment 5: Using const in Function Parameters

Objective: Understand how to use const with function parameters.

Write a function that takes a constant integer as an argument and prints its value. Attempting to modify this parameter inside the function should result in an error.

```
#include<stdio.h>

void Pi_value(const float pi){

   printf("Value of pi=%.2f",pi);

     // pi=3.14;

    }

int main(){

    float a=2.89;

    Pi_value(a);

    return 0;


}
```

Assignment 6: Array of Constants

Objective: Learn how to declare and use arrays with const.

Create an array of constants representing days of the week. Print each day using a loop, ensuring that no modifications can be made to the array elements

```
#include<stdio.h>

int main(){
    char *const
days[]={"SUNDAY","MONDAY","TUESDAY","WEDNESDAY","THURSDAY","FRIDAY","SATURDAY"};

    // days[1]="today";


    for(int i=0;i<7;i++){
        printf("%s\n",days[i]);
    }
}


/*
```

Assignment 7: Constant Expressions

Objective: Understand how constants can be used in expressions.

Write a program that uses constants in calculations, such as calculating the area of a circle using const.

```
#include<stdio.h>
int main(){
    const float pi=3.14;
    const float radius = 5;


    float area = pi*radius*radius;
    printf("Area=%.2f",area);


    // pi=2;
    // radius=9;
    // printf("Area=%.2f",area);


}
```

Assignment 8: Constant Variables in Loops

Objective: Learn how constants can be used within loops for fixed iterations.

Create a program that uses a constant variable to define the number of iterations in a loop, ensuring it cannot be modified during execution.

```
#include<stdio.h>

int main(){

   int const limit=10;

   for(int i=1;i<=limit;i++){

      printf("%d\n",i);

   }


   // limit=5;

   // for(int j=1;j<=limit;j++){

   //     printf("%d\n",j);

   // }


}
```

Assignment 9: Constant Global Variables

Objective: Explore global constants and their accessibility across functions.

Write a program that declares a global constant variable and accesses it from multiple functions without modifying its value

```
#include<stdio.h>

const float pi=3.14;


float circle_area(float radius){

   float area= pi*radius*radius;

   printf("Area of circle=%.2f\n",area);

}
```

```
float circle_circumference(float radius){
   float circumference=2*pi*radius;
   printf("Circumference of circle = %.2f\n",circumference);
}

int main(){
   float r=5;
   circle_area(r);
   circle_circumference(r);

   // pi=30;
   // circle_area(r);
}
```

10. Create a program that reverses the elements of an array. Prompt the user to enter values and print both the original and reversed arrays.

```
#include<stdio.h>
int main(){
   int array[10];
   printf("Enter 10 elements in the array");
   for(int i=0;i<10;i++){
      printf("%d th element: ",i);
      scanf("%d",&array[i]);
   }

   printf("array elemnts before reversing\n");
   for(int i=0;i<10;i++){
      printf("%d\n",array[i]);
   }


   printf("array elements after reversing\n");
   for(int i=9;i>=0;i--){
      printf("%d\n",array[i]);


   }
 }
```

11. Write a program that to find the maximum element in an array of integers. The program should prompt the user for input and display the maximum value.

```
#include<stdio.h>
 int main(){
   int array[10];
   printf("Enter 10 elements in the array");
   for(int i=0;i<10;i++){
      printf("%d th element: ",i);
      scanf("%d",&array[i]);
   }


   int max=array[0];
   for(int i=0;i<10;i++){
      if(array[i]>max){
         max=array[i];
      }
   }
   printf("Maximum value in the array= %d",max);


 }
```

12. Write a program that counts and displays how many times a specific integer appears in an array entered by the user.

```
#include<stdio.h>
 int main(){
   int array[10];
   int num,count=0;

   printf("Enter 10 elements in the array\n");
   for(int i=0;i<10;i++){
      printf("%d th element: ",i);
      scanf("%d",&array[i]);
   }

   printf("Enter the number to check the count");
   scanf("%d",&num);

    for(int i=0;i<10;i++){
      if(array[i]==num){
      count++;
      }
   }
  printf("Count of %d = %d",num,count);

 }
```

13.Program to print prime numbers

- **In this challnge you are going to create a program that will find all the prime numbers from 3-100**
- **There will no input to the program**
- **The Output will be each prime number separated by a space on a single line**
- **You will need to create an array that will store each prime number as it is generated**
- **You can hard code the first two prime numbers (2 and 3) in the prime array**

- **You should utilize loops to only find prime numbers upto 100 and a loop to print out the prime array**

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

int main() {
    int primes[50];
    int count = 2;
    primes[0] = 2;
    primes[1] = 3;

    for (int num = 4; num <= 100; num++) {
        bool isPrime = true;

        for (int i = 0; i < count; i++) {
            if (primes[i] * primes[i] > num) break;
            if (num % primes[i] == 0) {
                isPrime = false;
                break;
            }
        }

        if (isPrime) {
            primes[count] = num;
            count++;
        }
    }

    for (int i = 0; i < count; i++) {
        printf("%d ", primes[i]);
    }
```

```
   printf("\n");


   return 0;
}
```

14.Weather calculation

- **In this challenge, you are to create a C program that uses a two-dimensional array in a weather program.**
- **This program will find the total rainfall for each year, the average yearly rainfall, and the average rainfall for each month**
- **Input will be a 2D array with hard-coded values for rainfall amounts for the past 5 years**
- **The array should have 5 rows and 12 columns rainfall amounts can be floating point numbers**

```
#include<stdio.h>
int main(){
   float rain_fall[5][12]={
      {7,7.1,7.2,7.9,7,5,7,5,7.5,7.4,7.0,7.4},
      {7.9,7.1,7.2,7.9,7.7,5,7,5.9,7.5,7.4,7.0,7.4},
      {7,7.1,7.2,7.9,7,5,7,5.7,7.5,7.4,7.0,7.4},
      {7,7.1,7.2,7.7,7,5,7,5.7,7,7.4,7.0,7},
      {7,7.1,7.9,7.9,7,5,7.8,5.7,7.5,7.4,7.0,7.6},
   };
   int i,j;
   float total=0,year_average,month_average;
   int year[5]={2010,2011,2012,2013,2014};
   char
month[12][4]={"JAN","FEB","MAR","APR","MAY","JUN","JUL","AUG","SEP","OCT","NOV","DEC"};


  //year sum -->  row sum;
   printf("YEAR\tRAINFALL\n");
   for(i=0;i<5;i++){
      float year_sum=0;
```

```
        for(j=0;j<12;j++){
            year_sum=year_sum+rain_fall[i][j];
        }
        year_average=year_sum/5;
        printf("%d\t%.2f\n",year[i],year_average);

    }
    printf("\n");
    // float total_average=year_average/5;
    printf("The yearly average is %.2f inches\n",total_average);



    printf("MONTHLY AVERAGE\n");
    printf("\n");

    for(j=0;j<12;j++){
        float month_sum=0;
        for(i=0;i<5;i++){
            month_sum=month_sum+rain_fall[i][j];

        }
        printf("%s\t",month[j]);
        float month_average=month_sum/5;
        printf(" %.2f\n",month_average);
    }


}
```