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("The value of Pi is: %f\n", PI);

    return 0;
}
```

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 a=10;
    int b=20;
    int const *ptr = &a;
    printf("%d\n",*ptr);
    ptr = &b;
    printf("%d",*ptr);
}
```

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;
    printf("%d\n",*ptr);
    // ptr = &b; error occurs //*ptr = & b
    *ptr =30;
    printf("%d",*ptr);
}
```

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;
    const int * const ptr = &a;
    printf("%d\n",*ptr);
    // *ptr = 15;
    // printf("%d\n",*ptr);
}
```

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 modify(const int num ){
    printf("the number is %d",num);
    // num = 10;
}

int main(){
    int a=10;
    modify(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(){
    int const * day[10]
={"sunday","monday","tuesday","wednesday","thursday","friday","saturday"};
    for(int i=0;i<7;i++){
        printf("%s\n",day[i]);
    }
    // day[0]="new day";
 }
```

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>


float const pi =3.14159 ;

int main() {
    const float  pi =3.14159 ;
    float radius, area;
    printf("Enter the radius of the circle: ");
    scanf("%f", &radius);
    area = pi * radius * radius;
    printf("The area of the circle with radius %.2f is: %.2f\n", radius, area);

    return 0;
}
```

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(){
    const int fixed = 6;
    for(int i=0;i<fixed;i++){
        printf("%d\n",i);
    }
    // fixed =10;
}
```

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 int variable =10;
void display(int a){
    int v = variable + a;
    printf("%d",v);
}

int main(){
    int n;
    printf("enter a vlaue");
    scanf("%d",&n);
    display(n);




}
```

Assignment 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 arr[10],n;
    printf("enter the storage");
    scanf("%d",&n);
    printf("enter the element to be stored");
    for(int i=0;i<n;i++){
        scanf("%d",&arr[i]);
    }
    printf("the original element\n");
     for(int i=0;i<n;i++){
        printf("%d",arr[i]);
    }
    printf("\n");
    printf("the reversed element\n");

    for(int i = n-1;i>=0;i--){
        printf("%d",arr[i]);
    }

}
```

Assignment 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 arr[10],n,max;
    printf("enter the storage");
    scanf("%d",&n);
    printf("enter the element to be stored");
    for(int i=0;i<n;i++){
        scanf("%d",&arr[i]);
    }
    max=arr[0];
```

```
    for(int i=0;i<n;i++){
        if(arr[i]>max){
            max = arr[i];
        }
    }
    printf("the maximum value is %d",max);

}
```

Assignment 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 arr[10],n,count=0;
    printf("enter the storage");
    scanf("%d",&n);
    printf("enter the element to be stored");
    for(int i=0;i<n;i++){
    scanf("%d",&arr[i]);
    }
    for(int i=0;i<n;i++){
        count =1;
        if(arr[i]!=-1){
          for(int j=i+1;j<n;j++){
            if(arr[i]==arr[j]){
                count++;
                arr[j]=-1;
            }
        }
        }
            printf("%d repeat  %d times \n", arr[i],count);


    }


}
```

Assignment 13

```
#include <stdio.h>

int main() {
```

```
    int primes[100] = {2, 3};
    int count = 2;
    int isPrime;

    for (int i = 4; i <= 100; i++) {
        isPrime = 1;
        for (int j = 0; j < count; j++) {
            if (i % primes[j] == 0) {
                isPrime = 0;
                break;
            }
        }
        if (isPrime==1) {
            primes[count] = i;
            count++;
        }

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

    return 0;
}
```

Assignment 14
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

```
#include<stdio.h>
int main(){
    int sum=0,k=0;
    float avgmonth[10];
    float averageyear[10];
    int total;
    int years[5] = {2010, 2011, 2012, 2013, 2014};
```

```
   char *months[12] = {"Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep", "Oct", "Nov",
"Dec"};
int A[5][12];
for(int i=0;i<5;i++){
   for(int j=0;j<12;j++){
      scanf("%d",&A[i][j]);
      // avgmonth[j]=avgmonth[j]+A[k++][i];
      sum =sum +A[i][j];
   }
   averageyear[i]=averageyear[i]+sum;
   sum = 0;
}
for(int i=0;i<5;i++)
{
   averageyear[i]=averageyear[i]/12;
   printf("%d\t %f\n",years[i],averageyear[i]);
   total = total+averageyear[i];
}
total = total/5;
printf("the yearly average is %d",total);
for(int j=0;j<12;j++){
   for(int i=0;i<5;i++){
      avgmonth[j]=avgmonth[j]+A[i][j];
}
avgmonth[j]=avgmonth[j]/5;
}
for(int i=0;i<12;i++){
   printf("\n%s\t %f",months[i],avgmonth[i]);

}


}
```

CLASS WORKS

CONST

```
#include<stdio.h>


int main(){
    int const a =50;
    printf("001 a =%d\n",a);
    int * p;
    p= &a;
    *p=80;
    printf("002 a =%d\n",a);

}
```

ARRAY

```
#include<stdio.h>
int main(){
    int A[5];
    printf("size of the int %d\n",sizeof(int));
    printf("size of the array A =%d\n",sizeof(A));
    printf("A = %p\n",A); // address of array in first index
    for(int i=0;i<=4;i++){
        printf("A = %p\n",A+i);
        //(A+i)= base address of array + (index value * size of the data)
    }
}


#include<stdio.h>
int main(){
    int A[5];
    printf("enter the element in the array");
    for(int i=0;i<5;i++){
        scanf("%d",&A[i]);
```

```
    }
    for(int j =0;j<5;j++){
        printf("A[%d]=%d\n",j,A[j]);
    }
    return 0;
}


#include<stdio.h>
int main(){

     int temp[10],sum =0, average =0;
     printf("enter the grades");
     for(int i=0;i<10;i++){
        scanf("%d",&temp[i]);
        sum = sum +temp[i];
     }
     average =sum /10;
     printf("the sum is %d",sum);
     printf("the value of average is %d",average);

}



#include<stdio.h>
int main(){
    int A[5] = {1,2,3};
    printf("enter the element in the array");

    for(int j =0;j<5;j++){
        printf("A[%d]=%d\n",j,A[j]);
    }
    return 0;
}


#include<stdio.h>
# define month 12
int main(){
    int day[month]={31,[1]=29,[4]=31,30,31,30,31,30,31,30,};

    int i;
    for(i=0;i<month;i++){
```

```
        printf("[%d]  %d\n",  i+1,day[i]);
    }


}

#include<stdio.h>
int main(){
    int arr[10] = {0,1,4,9,16};
    int i;
    for(i=1;i<3;i++){
        arr[i]=i*i;
    }
    for(i=0;i<10;i++){
        printf("%d = ",arr[i]);
    }
}
```