CONST

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>

float const pi=3.14;

int main(){

   printf("The value of pi is %f",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=50;

   int const *ptr = (int*)&a;

   printf("001The value in pointer is %d",*ptr);

   //*ptr=80;

   //printf("002The value in pointer is %d",*ptr);

   return 0;

}
```

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=50;

   int *const ptr = (int*)&a;
```

```
    printf("001The value in ptr is %d\n",*ptr);
    *ptr=80;
    printf("002The value in ptr is %d",*ptr);
    int b=10;
    //ptr=(int*)&b;
    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=50;
    int const *const ptr=(int*)&a;
    printf("The value in ptr is %d",*ptr);
    //*ptr=80;
    //int b=10;
    //ptr=(int*)&b;
    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 fun(const int num){
    printf("%d",num);
    //num=20;
}
```

```
int main(){
   int const num=10;
   fun(num);
   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 *arr[]={"Monday","Tuesday","Wednesday","Thursday","Friday","Saturday","Sunday"};
   for(int i=0;i<=6;i++){
      printf("%s\n",arr[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(){
   float const pi=3.14;
   float area;
   int r=3;
   area=pi*r*r;
   printf("Area is %f",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 n=5;

   int i;

   for(i=1;i<=n;i++){

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

      //n+=2;

   }

}
```

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>

float const pi=3.14;

float area(float pi,int r){

   float a=pi*r*r;

   printf("Area is %f\n",a);

}

float perimeter(float pi,int r){

   float p=2*pi*r;

   printf("Perimeter is %f",p);

}

int main(){

   int r=3;

   area(pi,r);

   perimeter(pi,r);

   return 0;}
```

ARRAY:

User input an array

```
#include<stdio.h>
int main(){
   int a[5];
   printf("Enter the elements in the array a \n");
   for(int i=0;i<5;i++){
      scanf("%d",&a[i]);
      printf("\n");
   }
   for(int j=0;j<5;j++){
      printf("a[%d]=%d\n",j,a[j]);
   }
   return 0;
}
```

Average and sum

```
#include<stdio.h>
int main(){
   int grades[10];
   int count=10;
   int sum=0;
   float average;
   printf("Enter 10 grades :\n");
   for(int i=0;i<count;i++){
      scanf("%d",&grades[i]);
      sum+=grades[i];
   }
   printf("The sum of grades is %d",sum);
   average=(float)sum/count;
   printf("Average is %f",average);
   return 0;}
```

Month

```
#include<stdio.h>
#define MONTHS 12
int main(){
   int days[MONTHS]={31,28,31,30,31,30,31,31,30,31,30,31};
   int index;
   for(index=0;index<MONTHS;index++)
      printf("Month %d has %2d days.\n",index+1,days[index]);
   return 0;
}
```

Using designated initializer

```
#include<stdio.h>
#define MONTHS 12
int main(){
   int days[MONTHS]={31,28,[4]=31,30,31,[1]=29};
   int i;
   for(i=0;i<MONTHS;i++){
      printf("%2d %d\n",i+1,days[i]);
   }
   return 0;
}
```

Initializing all elements to the same value

```
#include<stdio.h>
int main(){
   int arr[10]={0,1,4,9,16};
   int i;
   for(i=5;i<10;i++){
      arr[i]=i*i;
   }
   for(i=0;i<10;i++){
```

```
        printf("arr[%i] = %i\n",i,arr[i]);
    }
    return 0;
}
```

Prime numbers

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[5];
    printf("Enter the values of array: ");
    for(int i=0;i<5;i++){
        scanf("%d",&arr[i]);
        printf("\n");
    }
    printf("The original array is : \n");
    for(int i=0;i<5;i++){
        printf("%d ",arr[i]);
    }
    printf("\n");
    printf("The reversed array is: \n");
    for(int i=4;i>=0;i--){
        printf("%d ",arr[i]);
    }
}
```

2. 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[5];
```

```
  printf("Enter the values of array: ");
 for(int i=0;i<5;i++){
    scanf("%d",&arr[i]);
    printf("\n");
 }
 int max=arr[0];
 for(int i=1;i<5;i++){
    if(arr[i]>max){
       max=arr[i];
    }
 }
 printf("Maximum element in the array is %d",max);
 return 0;

}
```

3. 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[5];
 int num,count=0;
  printf("Enter the values of array: ");
 for(int i=0;i<5;i++){
    scanf("%d",&arr[i]);
    printf("\n");
 }
 printf("Enter the number to check its count: ");
 scanf("%d",&num);
 for(int i=0;i<5;i++){
    if(arr[i]==num){
       count++;
```

```
        }
    }
    if(count>0){
        printf("%d appears %d times in the array",num,count);
    }
    else{
        printf("%d is not present in the array",num);
    }
    return 0;
}
#include <stdio.h>

int main() {

    int arr[5];
    printf("Enter elements of the array:\n");
    for (int i = 0; i < 5; i++) {
        scanf("%d", &arr[i]);
    }
    int counted[5];
    for (int i = 0; i < 5; i++) {
        counted[i] = 0;

    for (int i = 0; i < 5; i++) {
        if (counted[i] == 0) {
            int count = 1;
            for (int j = i + 1; j < 5; j++) {
                if (arr[i] == arr[j]) {
                    count++;
                    counted[j] = 1;
                }
```

```
            }
            printf("The integer %d appears %d times in the array.\n", arr[i], count);
            counted[i] = 1;
        }
    }

    return 0;
}
}
```

MATRIX

```
#include<stdio.h>
int main(){
    int A[4][5]={
        {1,2,3,4,5},
        {6,7,8,9,10},
        {11,12,13,14,15},
        {16,17,18,19,20}
    };
for(int j=0;j<4;j++){
    for(int k=0;k<5;k++){
        printf("%d ",A[j][k]);
    }
    printf("\n");
}
}
```

Designated initializer

```
#include<stdio.h>
int main(){
    int A[3][3]={[0][0]=1,[1][1]=1,[2][2]=1};
for(int j=0;j<3;j++){
    for(int k=0;k<3;k++){
```

```
        printf("%d ",A[j][k]);
    }
    printf("\n");
}
}
```

3-dimensional array

```
#include<stdio.h>
int main(){
    int sum=0;
    int num[2][2][2]={
        {
            {1,2},
            {3,4}
        },
        {
            {5,6},
            {7,8}
        }

    };

    for(int i=0;i<2;i++){
        for(int j=0;j<2;j++){
            for(int k=0;k<2;k++){
                sum+=num[i][j][k];
            }
        }
    }
    printf("Sum of all the elements in a 3 dimensional array is %d",sum);
    return 0;
}
```

Create a c program that uses 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

Example output:

Year rainfall (inches)

2010 32.4

2011 37.9

2012 49.8

2013 44.0

2014 32.9

The yearly average is 39.4 inches.

Monthly averages:

Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
7.3	7.3	4.9	3.0	2.3	1.2	0.3	0.5	1.7	3.6	6.7	

```
#include<stdio.h>
int main(){
   float rainfall[5][12]={
      {7.3, 7.3, 4.9, 3.0, 2.3, 1.2, 0.3, 0.5, 1.7, 3.6, 6.7, 3.9},
      {7.3, 7.3, 4.9, 3.0, 2.3, 1.2, 0.3, 0.5, 1.7, 3.6, 6.7, 3.9},
      {9.0, 7.5, 5.2, 3.5, 2.6, 1.5, 0.6, 0.7, 2.3, 4.0, 7.2, 4.0},
      {7.2, 8.0, 5.5, 3.3, 2.1, 1.0, 0.2, 0.4, 1.5, 3.2, 6.8, 3.8},
      {7.8, 6.9, 4.8, 2.8, 2.0, 1.1, 0.3, 0.5, 1.8, 3.4, 6.5, 4.0}
   };
   float total[5]={0};
   float month_avg[12]={0};
   float totalrainfall=0;
   float avgrainfall;
   char *months[12]={"Jan","Feb","Mar","Apr","May","Jun","Jul","Aug","Sep","Oct","Nov","Dec"};
   int years[5]={2010,2011,2012,2013,2014};
```

```
    for(int year=0;year<5;year++){
        total[year]=0;
        for(int month=0;month<12;month++){
            total[year]+=rainfall[year][month];
        }
        totalrainfall+=total[year];
    }
    for(int month=0;month<12;month++){
        for(int year=0;year<5;year++){
            month_avg[month]+=rainfall[year][month];
        }
        month_avg[month]/=5;
    }
    printf("YEAR  RAINFALL (inches)\n");
    for(int year=0;year<5;year++){
        printf("%d: %.1f inches\n",years[year],total[year]);
    }
    printf("\nThe yearly average is %.1f inches.\n",totalrainfall/5);
    printf("\nMONTHLY AVERAGES:\n");
    for(int month=0;month<12;month++){
        printf("%s ",months[month]);
    }
    printf("\n");
    for(int month=0;month<12;month++){
        printf("%.1f ",month_avg[month]);
    }
    return 0;
}
```