PHASE 1 DAY 7

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);

   // Pi = 3.14159; //compilation error

    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 num = 10;

    const int *ptr = &num;

    printf("Value is: %d\n", *ptr);

    // *ptr = 20;//compilation error

    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 x = 10;
```

```
    int y = 20;

    int *const ptr = &x;

    printf("Value pointed by ptr is: %d\n", *ptr);

    // ptr = &y;//compilation error

    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 num = 10;

    const int *const ptr = &num;

    printf("Value pointed by ptr is: %d\n", *ptr);

    // *ptr = 20;

    // ptr = &value;

    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 printnum(const int num)

 {

    printf("Number is: %d\n",num);

    // num = 20;

}

int main()
```

```
{
    int x = 10;
    printnum(x);
    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 the array elements cannot be modified.

```
#include <stdio.h>

int main()

{
    const char *week[] = {"Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"};
    for (int i = 0; i < 7; i++) {
        printf("%s\n",week[i]);
    }
    // week[0] = “Nanda";
    return 0;
}
```

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;
    float r = 5;
    float area = Pi * r * r;
    printf("Area of the circle: %f\n", 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 n = 5;

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

        printf("Iteration %d\n", i + 1);

    }

    // n= 10;//compilation error;trying to change value

    return 0;

}
```

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 x= 100;

void print()

 {

    printf("001 x : %d\n",x);

}

int main()

 {

    print();

    printf("002 x: %d\n",x);

    // x = 200;

    return 0;

}
```

ARRAYS

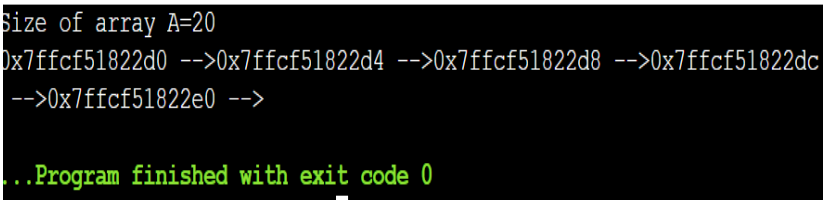
Memory allocation

1.#include<stdio.h>

int main()

{

```
    int A[5];

                printf("Size of array A=%d\n",sizeof(A));

    for(int i=0;i<=4;i++)

    {

        printf("%p -->",(A+i));

        //base address+(index*size of data type)

    }

    return 0;

}
```



2. Average of grades of 10 students

```
#include<stdio.h>

int main()

{

    int grades[10];

    int count=10;

    long sum=0;

    float avg=0.0f;

    printf("Enter the 10 grades:\n ");

    for(int i=0;i<10;i++)

    {

        printf("%2u>",i+1);

        scanf("%d",&grades[i]);

        sum+=grades[i];

    }
```

```
    avg=(float)sum/count;
    printf("\n Average of ten grades =%.2f\n",avg);
    return 0;
}
```

3. Print prime numbers from 3 to 100

```
#include<stdio.h>
int main()
{
    int prime[50]={3};
    int k=1;
    for(int i=4;i<100;i++)
    {
        int flag=0;
      for(int j=2;j<i;j++)
      {
        if(0==i%j)
        {
            flag=1;
            break;
        }
      }
        if(flag==0)
        {
            prime[k]=i;
            k++;
        }
    }
   printf("Prime nos between 3 and 100:");
    for(int i=0;i<k;i++)
    {
```

```
Prime nos between 3 and 100:3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97
```

```
    printf("%d ",prime[i]);
    }
    return 0;
}
```

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

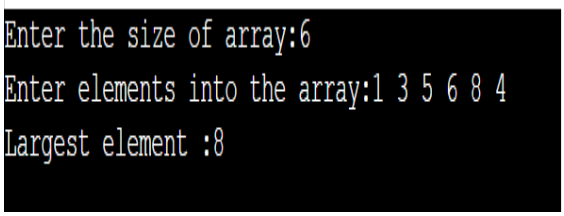
```
#include<stdio.h>
int main()
{
    int size;
    printf("Enter the size of array:");
    scanf("%d",&size);

    int A[size];

    printf("Enter elements into the array:");
    for(int i=0;i<size;i++)
    {
    scanf("%d",&A[i]);
    }

   printf("Original Array:");
    for(int i=0;i<size;i++)
    {
    printf("%d ",A[i]);
    }
    printf("\n");

    printf("Reversed Array:");
    for(int i=size-1;i>=0;i--)
```

```
Enter the size of array:4
Enter elements into the array:1 2 3 4
Original Array:1 2 3 4
Reversed Array:4 3 2 1
```

```
    {
    printf("%d ",A[i]);
    }
    return 0;
}
```

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

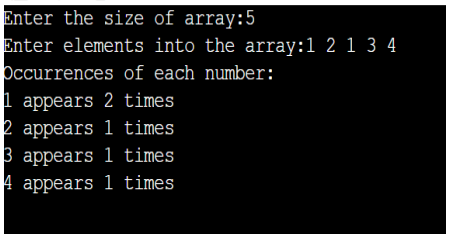
```
#include<stdio.h>
int main()

{
    int size;
    printf("Enter the size of array:");
    scanf("%d",&size);
       int A[size];
       printf("Enter elements into the array:");
    for(int i=0;i<size;i++)
    {
    scanf("%d",&A[i]);
    }
       int largest =A[0];
    for(int j=0;j<size;j++)
    {
       if(A[j]>largest)
       {
          largest=A[j];
       }
    }
```



```
    printf("Largest element :");
    printf("%d ",largest);
    return 0;
}
```

6. 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 size,count=0;
    printf("Enter the size of array:");
    scanf("%d",&size);
    int A[size];
    printf("Enter elements into the array:");
    for(int i=0;i<size;i++)
    {
    scanf("%d",&A[i]);
    }
    printf("Occurrences of each number:\n");
    for (int i = 0; i < size; i++)
    {
        int count = 1;
        int alreadyCounted = 0;
        for (int k = 0; k < i; k++)
        {
            if (A[i] == A[k])
            {
                alreadyCounted = 1;
                break;
            }
```



```
        }
        if (alreadyCounted)
        continue;
        for (int j = i + 1; j < size; j++)
        {
            if (A[i] == A[j])
            {
                count++;
            }
        }
                printf("%d appears %d times\n", A[i], count);
    }
    return 0;
}
```

7.3D array Sum

```
#include<stdio.h>
int sum=0;
int main()
{
   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=%d",sum);
  return 0;
}
```

Output:

Sum=36

8.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>
void main() {
```

```
    float rain_data[5][12] =
{
        {4.5, 5.0, 6.2, 3.1, 4.8, 5.3, 6.1, 4.0, 3.7, 4.9, 5.3, 4.6},
        {4.2, 4.6, 6.0, 3.8, 4.9, 5.0, 5.8, 4.6, 3.9, 4.7, 4.9, 5.1},
        {3.8, 4.9, 5.7, 3.3, 4.6, 5.5, 6.0, 4.3, 3.6, 4.8, 5.1, 4.8},
        {4.0, 4.8, 6.3, 3.6, 4.7, 5.2, 5.9, 4.4, 3.8, 4.6, 5.0, 4.9},
        {4.3, 5.1, 6.1, 3.5, 4.5, 5.1, 5.7, 4.1, 3.9, 4.5, 5.2, 4.7}
    };

    float yearly_total_rain[5] = {0};
    float yearly_avg_rain[5] = {0};
    float monthly_avg_rain[12] = {0};

    char months[12][4] = {"Jan", "Feb", "Mar", "Apr", "May", "Jun",
                          "Jul", "Aug", "Sep", "Oct", "Nov", "Dec"};


    for (int i = 0; i < 5; i++) {
        for (int j = 0; j < 12; j++) {
            yearly_total_rain[i] += rain_data[i][j];
            monthly_avg_rain[j] += rain_data[i][j];
        }
        yearly_avg_rain[i] = yearly_total_rain[i] / 12;
    }


    for (int j = 0; j < 12; j++) {
        monthly_avg_rain[j] /= 5;
    }
```

```
Total rainfall for each year:
Year 2020: 57.50
Year 2021: 57.50
Year 2022: 56.40
Year 2023: 57.20
Year 2024: 56.70

Average rainfall for each year:
Year 2020: 4.79
Year 2021: 4.79
Year 2022: 4.70
Year 2023: 4.77
Year 2024: 4.72

Average monthly rainfall over 5 years:
Jan: 4.16
Feb: 4.88
Mar: 6.06
Apr: 3.46
May: 4.70
Jun: 5.22
Jul: 5.90
Aug: 4.28
Sep: 3.78
Oct: 4.70
Nov: 5.10
Dec: 4.82
```

```
    printf("Total rainfall for each year:\n");
    for (int i = 0; i < 5; i++) {
        printf("Year 202%d: %.2f\n", i, yearly_total_rain[i]);
    }

    printf("\nAverage rainfall for each year:\n");
    for (int i = 0; i < 5; i++) {
printf("Year 202%d: %.2f\n", i, yearly_avg_rain[i]);
}
printf("\nAverage monthly rainfall over 5 years:\n");
for (int j = 0; j < 12; j++) {
printf("%s: %.2f\n", months[j], monthly_avg_rain[j]);
}
}
```