

### Assignment 1: Constant Variable Declaration

Objective: Learn to declare and initialise 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 int pi = 3.14;

    printf("Value of pi: %d\n", pi);

    pi = 6;

    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 n = 10;

    const int *p = &n;

    printf("Value of num through pointer: %d\n", *p);
}
```

```

    *p = 20;

    n = 20;

    printf("Updated value of num: %d\n", n);

    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 n1 = 10;

    int n2 = 20;

    int *const p = &n1;

    printf("001Value of n1:: %d\n", *p);

    *p = 15;

    printf("002Value of n1:: %d\n", *p);

    p = &n2;

    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 n1 = 10;  
  
    int n2 = 20;  
  
    const int *const p = &n1;  
  
    printf("001Value of n1:: %d\n", *p);  
  
    *p = 15;  
  
    printf("002Value of n1:: %d\n", *p);  
  
    p = &n2;  
  
    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 Constant(const int n) {

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

    n = 30;

}

int main() {

    int num = 10;

    Constant(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() {

    int i;

    char *const a[7] = {"Sunday", "Monday", "Tuesday",
"Wednesday", "Thursday", "Friday", "Saturday"};

    for (i = 0; i < 7; i++) {

        printf("%s\n", a[i]);

    }

}

```

```
a[0] = "abcd";

printf("%d",a[0]);


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 int pi = 3.14;

    int a,r=4;

    a=pi*r*r;

    printf("Area::%d",a);


    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 i = 0;  
  
    int n=10;  
  
    for(i=0;i<n;i++){  
  
        printf("%d\n",i);  
  
    }  
  
    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 a = 30;  
  
int main() {  
  
    printf("001Value of a: %d\n", a);  
  
    a = 30;
```

```

printf("002Value of a: %d\n", a);

return 0;
}

```

11. Create a program that reverses the elements of an array. Prompt the user to enter values and print both the original and reversed arrays. 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 n,i;

    printf("Enter how many elements in array:");

    scanf("%d",&n);

    int a[n];

    printf("\nEnter %d elements in array::",n);

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

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

    }

    printf("\nOriginal array a[%d]=",n);

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

        printf("%d\t",a[i]);
    }
}

```

```

    }

    printf("\nReversed array a[%d]=",n);

    for(i=n-1;i>=0;i--){

        printf("%d\t",a[i]);

    }

}

```

12. 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 a[5];

    printf("Enter 5 values into array::");

    for(int i=0;i<5;i++){

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

    }

    int largest=a[0];

    for(int i=0;i<5;i++){

        if(a[i]>largest){

            largest=a[i];

        }

        else{

            continue;

        }

    }

}

```



```
printf("\nLargest is %d",largest);  
}
```

13. 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 n,c;  
  
    int count=0;  
  
    printf("Enter size of array::");  
  
    scanf("%d",&n);  
  
    int a[n];  
  
    printf("\nEnter %d elements into array::",n);  
  
    for(int i=0;i<n;i++){  
  
        scanf("%d",&a[i]);  
  
    }  
  
    printf("\nEnter element to find count::");  
  
    scanf("%d",&c);  
  
    for(int i=0;i<n;i++){  
  
        if(a[i]==c) {  
  
            count++;  
  
        }  
  
        else{  
  
            continue;  
  
        }  
  
    }  
  
}
```

```
}

printf("\n%d has appeared %d times",c,count);

}
```

## 14.Weather

```
#include <stdio.h>

int main() {

    int a[5][12];

    int i, j;

    int year, avg_year = 0;

    printf("Enter Starting year: ");

    scanf("%d", &year);

    printf("Enter the rainfall into 5x12 Matrix (where each row is a  
year and each column is a month):\n");

    for (i = 0; i < 5; i++) {

        for (j = 0; j < 12; j++) {

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

        }

    }

    int row_sum[5] = {0};

    for (i = 0; i < 5; i++) {
```

```
        for (j = 0; j < 12; j++) {

            row_sum[i] += a[i][j];

        }

    }

    for (i = 0; i < 5; i++) {

        avg_year += row_sum[i];

    }

    avg_year = avg_year / 5;

    int column_sum[12] = {0};

    for (j = 0; j < 12; j++) {

        for (i = 0; i < 5; i++) {

            column_sum[j] += a[i][j];

        }

    }

    int avg_month[12];

    for (i = 0; i < 12; i++) {

        avg_month[i] = column_sum[i] / 5;

    }

    printf("\nYEAR\t\tRAINFALL (inches)");
```

```
for (i = 0; i < 5; i++) {

    printf("\n%d\t\t%d", year + i, row_sum[i]);

}


printf("\nThe Yearly Average is %d inches", avg_year);


printf("\nMONTHLY AVERAGE:\n");

printf("JAN\tFEB\tMAR\tAPR\tMAY\tJUN\tJUL\tAUG\tSEP\tOCT\tNOV\tDEC\n");

for (i = 0; i < 12; i++) {

    printf("%d\t", avg_month[i]);

}


return 0;

}
```