

DAY 7-DAILY ASSIGNMENTS

ANSU MARIUM SHIBU

14-11-2024

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("pi=%f\n",pi);
    pi=3.14567;
    return 0;
}
```

```
PS D:\c progrms coding> gcc constassi1.c
constassi1.c: In function 'main':
constassi1.c:6:8: error: assignment of read-only variable 'pi'
   6 |     pi=3.14567;
     |     ^
```

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(){
    const int a=50;
    const int* ptr=&a;
    *ptr=500;

    printf("value is:%d\n",*ptr);
    return 0;
}
```

```
PS D:\c progrms coding> gcc constassi2.c
constassi2.c: In function 'main':
constassi2.c:6:10: error: assignment of read-only location '*ptr'
    6 |     *ptr=500;
      |     ^
```

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 b=90;
    int* const ptr=&a;
    printf("Value pointed : %d\n", *ptr)
    *ptr=30;
    printf("value is modifi:%d\n",*ptr);
    return 0;
}
```

```
PS D:\c progrms coding> gcc constassi3.c
PS D:\c progrms coding> ./a
Value pointed : 50
value is:30
```

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.

```
constassi4.c > main()
1  #include<stdio.h>
2  int main(){
3      const int a=90;
4      const int *const ptr=&a;
5      printf("value pointe:%d\n",*ptr);
6
7      *ptr=20;
8      ptr=&a;
9
10
11 }
```

```
PS D:\c progrms coding> gcc constassi4.c
constassi4.c: In function 'main':
constassi4.c:7:9: error: assignment of read
-only location '*(const int *)ptr'
    7 |     *ptr=20;
      |         ^
constassi4.c:8:9: error: assignment of read
-only variable 'ptr'
    8 |     ptr=&a;
      |         ^
```

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>

int main(){
    const int a=10;
    printf("value:%d\n",a);

    a=80;
    return 0;
}
```

```
PS D:\c progrms coding> gcc constassi5.c
constassi5.c: In function 'main':
constassi5.c:7:6: error: assignment of read-only variable 'a'
    7 |     a=80;
      |     ^
```

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

```
PS D:\c progrms coding> gcc constassi6.c
PS D:\c progrms coding> ./a
Sunday
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
```

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 r=4.0;
    const float area=pi*r*r;
    printf("area=%f\n",area);
    return 0;
}
```

```
PS D:\c progrms coding> gcc constassi7.c
PS D:\c progrms coding> ./a
area=50.240002
```

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 iter=5;

    for(int i=0;i<iter;i++)
    {
        printf("iterations=%d\n",i+1);
    }
    return 0;
}
```

```
PS D:\c progrms coding> gcc constassi8.c
PS D:\c progrms coding> ./a
iterations=1
iterations=2
iterations=3
iterations=4
iterations=5
PS D:\c progrms coding>
```

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 gb=100;

int main(){

    printf("gloabl varia:%d\n",gb);
    return 0;
}
```

```
PS D:\c progrms coding> gcc constassi9.c
PS D:\c progrms coding> ./a
gloabl varia:100
```

10. •In this challenge, you are going to create a program that will find all the prime numbers from 3-100

- there will be 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 primes array
- You should utilize loops to only find prime numbers up to 100 and a loop to print out the primes array

```

#include<stdio.h>

int main(){
    int prime[100];
    int index=2;
    prime[0]=2;
    prime[1]=3;

    for(int i=5;i<=100;i+=2){
        int j;
        for (j=1;j<index;j++){
            {
                if(i%prime[j]==0){
                    break;
                }
            }
            if(j==index){
                prime[index++]=i;
            }
        }
        for(int i=0;i< index;i++)
        {
            printf("%d\n",prime[i]);
        }
    }
}

```

PS D:\c progrms coding> gcc arraytas.c

PS D:\c progrms coding> ./a

2
 3
 5
 7
 11
 13
 17
 19
 23
 29
 31
 37
 41
 43
 47
 53
 59

```
59
61
67
71
73
79
73
79
83
79
83
89
97
83
89
97
89
97
```

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.


```

#include<stdio.h>

int main(){
    int n;
    printf("enter number of ele:\n");
    scanf("%d",&n);

    int arr[n];

    printf("enter numbers:\n");
    for(int i=0;i<n;i++)
    {
        scanf("%d",&arr[i]);
    }

    printf("enter original arr:\n");
    for(int i=0;i<n;i++)
    {
        printf("%d",arr[i]);
    }
    printf("\n");

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

```

```

PS D:\c progrms coding> ./a
PS D:\c progrms coding> ./a
enter number of ele:
4
enter numbers:
1
2
1
2
2
3
4
enter original arr:
enter original arr:
1234
reverse arr:
4321
PS D:\c progrms coding>

```

12. 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 n;
    printf("enter number of ele:\n");
    scanf("%d",&n);

    int arr[n];

    printf("enter numbers:\n");
    for(int i=0;i<n;i++)
    {
        scanf("%d",&arr[i]);
    }

    int max=arr[0];
    for(int i=1;i<n;i++)
    {
        if(arr[i]>max)
        {
            max=arr[i];
        }
    }
    printf("The maximum element : %d\n", max);
}
```

```
PS D:\c progrms coding> gcc arrayassi2.c
PS D:\c progrms coding> ./a
enter number of ele:
5
enter numbers:
2
5
7
1
8
The maximum element : 8
```

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

```

1  #include <stdio.h>
2
3  int main() {
4      int n, search, count = 0;
5
6      printf("Enter number of elements: ");
7      scanf("%d", &n);
8
9      int arr[n];
10
11     printf("Enter numbers:\n");
12     for (int i = 0; i < n; i++) {
13         scanf("%d", &arr[i]);
14     }
15
16     printf("Enter integer to search: ");
17     scanf("%d", &search);
18
19     for (int i = 0; i < n; i++) {
20         if (arr[i] == search) {
21             count++;
22         }
23     }
24
25     if (count > 0) {
26         printf("The number %d appears %d time(s) in the array.\n", search, count);
27     } else {
28         printf("The number %d does not appear in the array.\n", search);
29     }
30 }

```

```

PS D:\c progrms coding> gcc arrayassi3.c
PS D:\c progrms coding> ./a
Enter number of elements: 6
Enter numbers:
2
2
2
4
1
6
4
Enter integer to search: 2
1
6
4
Enter integer to search: 2
4
Enter integer to search: 2
Enter integer to search: 2
The number 2 appears 2 time(s) in the array

```

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 rainfall amounts can be floating point numbers .simple way program

```

#include <stdio.h>

int main() {
    float rainfall[5][12] = {
        {7.3, 7.3, 4.9, 3.0, 2.3, 0.6, 1.2, 0.3, 0.5, 1.7, 3.6, 6.7},
        {7.5, 7.4, 5.1, 3.1, 2.5, 1.1, 1.3, 0.2, 0.6, 1.8, 3.8, 6.9},
        {8.0, 7.6, 5.5, 3.5, 2.8, 1.3, 1.5, 0.4, 0.7, 2.0, 4.0, 7.2},
        {6.9, 7.1, 4.8, 3.2, 2.6, 1.0, 1.4, 0.3, 0.8, 2.1, 3.7, 6.5},
        {7.2, 7.0, 5.3, 3.3, 2.7, 1.2, 1.6, 0.4, 0.9, 1.9, 3.9, 6.8}
    };

    float yearlyRainfall[5];
    float monthlyAverage[12] = {0};

    for (int i = 0; i < 5; i++) {
        yearlyRainfall[i] = 0;
        for (int j = 0; j < 12; j++) {
            yearlyRainfall[i] += rainfall[i][j];
            monthlyAverage[j] += rainfall[i][j];
        }
    }

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

    printf("YEAR    RAINFALL (inches)\n");
    for (int i = 0; i < 5; i++) {
        printf("%d    %.1f\n", 2010 + i, yearlyRainfall[i]);
    }
}

```

```
printf("YEAR    RAINFALL (inches)\n");
for (int i = 0; i < 5; i++) {
    printf("%d    %.1f\n", 2010 + i, yearlyRainfall[i]);
}

float totalRainfall = 0;
for (int i = 0; i < 5; i++) {
    totalRainfall += yearlyRainfall[i];
}

printf("\nThe yearly average is %.1f inches.\n", totalRainfall / 5);

printf("\nMONTHLY AVERAGES:\n");
for (int j = 0; j < 12; j++) {
    switch (j) {
        case 0:
            printf("Jan\t");
            break;
        case 1:
            printf("Feb\t");
            break;
        case 2:
            printf("Mar\t");
            break;
        case 3:
            printf("Apr\t");
            break;
    }
}
```

```
2         printf("Apr\t");
3         break;
4     case 4:
5         printf("May\t");
6         break;
7     case 5:
8         printf("Jun\t");
9         break;
10    case 6:
11        printf("Jul\t");
12        break;
13    case 7:
14        printf("Aug\t");
15        break;
16    case 8:
17        printf("Sep\t");
18        break;
19    case 9:
20        printf("Oct\t");
21        break;
22    case 10:
23        printf("Nov\t");
24        break;
25    case 11:
26        printf("Dec\t");
27        break;
```

```

        break;
    case 7:
        printf("Aug\t");
        break;
    case 8:
        printf("Sep\t");
        break;
    case 9:
        printf("Oct\t");
        break;
    case 10:
        printf("Nov\t");
        break;
    case 11:
        printf("Dec\t");
        break;
    }
}

printf("\n");

for (int j = 0; j < 12; j++) {
    printf("%.1f\t", monthlyAverage[j]);
}
printf("\n");

```

PS D:\c\progrms coding> gcc twoassi1.c

PS D:\c\progrms coding> ./a

YEAR RAINFALL (inches)

2010 39.4

2011 41.3

2012 44.5

2013 40.4

2014 42.2

The yearly average is 41.6 inches.

MONTHLY AVERAGES:

Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
7.4	7.3	5.1	3.2	2.6	1.0	1.4	0.3	0.7	1.9	3.8	6.8

PS D:\c\progrms coding>

