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("001 pi is %f",pi);

  pi=34.0;

  printf("002 pi is %f ",pi);

}
```

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>

const int a=80;// using global scope;

int main()

{


   printf("001 a =%d \n",a);

   int *p;

   p=&a;

   *p=180;

   printf("002 a=%d \n",a);

}
```

```
PS D:\learning c\output> & .\'day7_ass2.exe'
001 a =80
```

```
uint8_t const *pdata = (uint8_t)0x4000000;
```

```
    *pdata=800;
```

```
d:\learning c\day7_ass2.c:14:11: error: assignment of read-only location '*pdata'
     *pdata=800;
           ^
```

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>

#include<stdint.h>


int main()

{

   uint8_t *const pdata=(uint8_t*)0x400000;

   pdata=(uint8_t*)0x500000;


}
```

```
d:\learning c\day7_ass3.c:7:10: error: assignment of read-only variable 'pdata'
     pdata=(uint8_t*)0x500000;
           ^
```

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>

#include<stdint.h>


int main()

{
```

```
    uint8_t const *const pdata=(uint8_t*)0x400000;
    pdata=(uint8_t*)0x500000;
    *pdata=90;

}
```

```
d:\learning c\day7_ass4.c:7:10: error: assignment of read-only variable 'pdata'
     pdata=(uint8_t*)0x500000;
          ^
d:\learning c\day7_ass4.c:8:11: error: assignment of read-only location '*pdata'
     *pdata=90;
           ^
```

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 value(const int num)
   {
      printf("value of num is %d",num);
      num=num+1;
   }
int main()
{
   int num=76;
   value(num);

}
```

```
d:\learning c\day7_ass5.c: In function 'value':
d:\learning c\day7_ass5.c:5:12: error: assignment of read-only parameter 'num'
          num=num+1;
             ^
```

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]);
    }



    daysOfWeek[0] = "Funday";


    return 0;
}
```

```
PS D:\learning c\output> cd 'd:\learning c\output'
PS D:\learning c\output> & .\'day7_ass6.exe'
Sunday
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
PS D:\learning c\output>
```

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 pi=3.14;

   float radius=2.0;

   float area=pi*radius*radius;

   printf("area is %f",area);


}
```

```
PS D:\learning c\output> & .\'day7_ass7.exe'
area is 12.000000
```

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("no.of iterations %d  \n ",i+1);
```

```
    }
}
```

```
PS D:\learning c\output> cd 'd:\learning c\output'
PS D:\learning c\output> & .\'day7_ass8.exe'
no.of iterations 1
 no.of iterations 2
 no.of iterations 3
 no.of iterations 4
 no.of iterations 5
```

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 n=5;

void num()

{

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

}

void area()

{


   printf("Area of square is %d * %d = %d \n", n, n, n * n);

}

int main()

{

   num();

   area();

}
```

```
PS D:\learning c\output> cd 'd:\learning c\output'
PS D:\learning c\output> & .\'day7_ass9.exe'
number is 5
Area of square is 5 * 5 = 25
PS D:\learning c\output>
```

10.

```
#include <stdio.h>
#include <math.h>

int main() {
    int prime[100];
    int count = 0;

    for (int n = 3; n < 100; n++) {
        int isPrime = 1;

        for (int i = 2; i <= sqrt(n); i++)
        {
            if (n % i == 0) {
                isPrime = 0;
                break;
            }
        }

        if (isPrime) {
            prime[count] = n;
            count++;
        }
    }
```

```
    printf("Prime numbers between 3 and 100 are:\n");
    for (int i = 0; i < count; i++) {
        printf("array values[%i]=%i \n", i, prime[i]);
    }
    printf("\n");

    return 0;
}
```

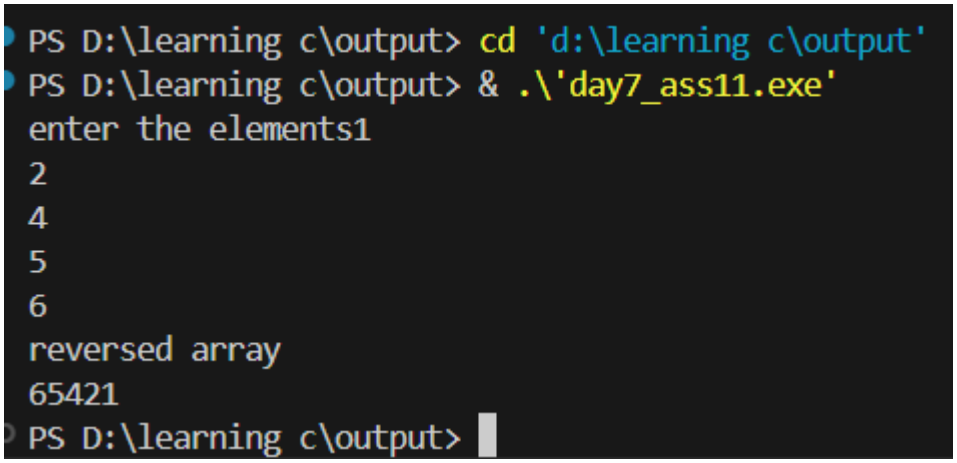
```
Prime numbers between 3 and 100 are:
array values[0]=3
array values[1]=5
array values[2]=7
array values[3]=11
array values[4]=13
array values[5]=17
array values[6]=19
array values[7]=23
array values[8]=29
array values[9]=31
array values[10]=37
array values[11]=41
array values[12]=43
array values[13]=47
array values[14]=53
array values[15]=59
array values[16]=61
array values[17]=67
array values[18]=71
array values[19]=73
array values[20]=79
array values[21]=83
array values[22]=89
array values[23]=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>
```

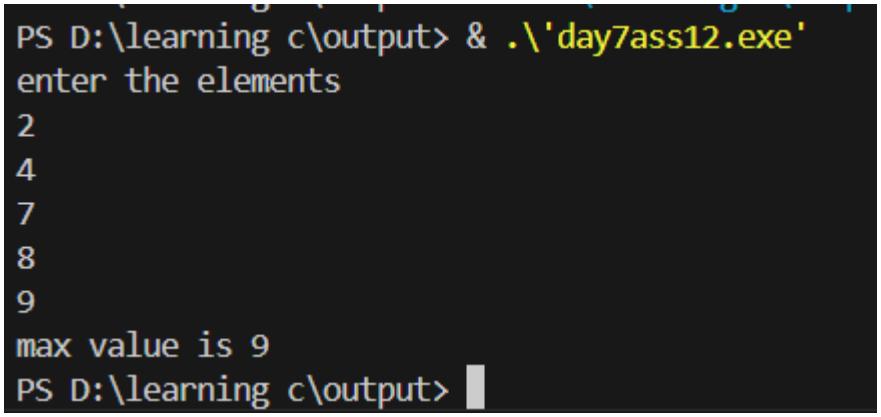
```
int main()
{
   int a[5];
   int start=0;
   int end=4;
   printf("enter the elements");
   for( int i=0;i<5;i++)
   {
     scanf("%d",&a[i]);
   }
   int rev[5];
   while(start<end)
   {
      int temp=a[start];
      a[start]=a[end];
      a[end]=temp;
      start++;
      end--;
   }
   printf("reversed array \n");
   for(int i=0;i<5;i++)
   {
      printf("%d",a[i]);
   }
}
```



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.*/

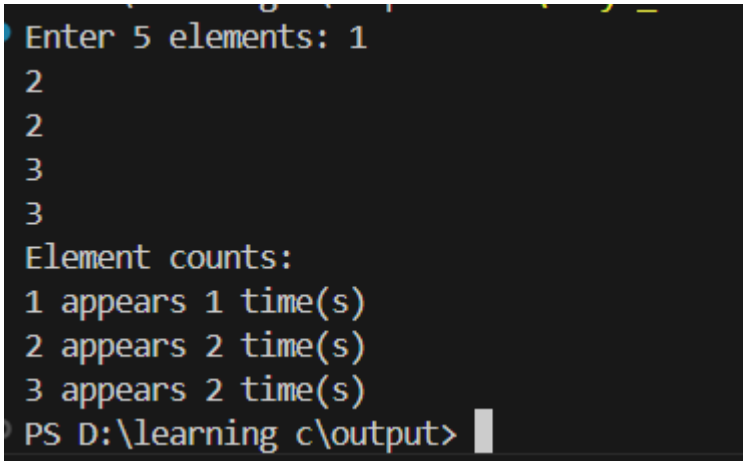
```
#include<stdio.h>
int main()
{
   int a[5];
   int i;
   printf("enter the elements \n");
   for(i=0;i<5;i++)
   {
      scanf("%d",&a[i]);
   }
   int max=a[0];
   for(i=0;i<5;i++)
   {
      if(a[i]>=max)
      {
         max=a[i];
      }
   }
   printf("max value is %d",max);

}
```



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 a[5], count[5];
   int i, j;
   int n = 5;



   printf("Enter 5 elements: ");
   for (i = 0; i < n; i++) {
      scanf("%d", &a[i]);
      count[i] = -1;
   }



   for (i = 0; i < n; i++) {
      int c = 1;
      for (j = i + 1; j < n; j++) {
         if (a[i] == a[j]) {
            c++;
            count[j] = 0;
         }
```

```
        }
        if (count[i] != 0) {
            count[i] = c;
        }
    }


    printf("Element counts:\n");
    for (i = 0; i < n; i++) {
        if (count[i] != 0) {
            printf("%d appears %d time(s)\n", a[i], count[i]);
        }
    }

    return 0;
}
```



14.

Assignment

Requirements

- 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

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  0.6  1.2  0.3  0.5  1.7  3.6  6.7
```

```
#include <stdio.h>


#define YEARS 5

#define MONTHS 12


int main() {

   float rainfall[YEARS][MONTHS] = {

      {7.3, 6.8, 6.1, 3.2, 0.0, 0.0, 0.2, 0.3, 0.7, 2.1, 5.4, 6.4},

      {4.3, 6.4, 6.4, 2.2, 2.1, 2.7, 1.7, 0.6, 1.8, 1.1, 5.6, 7.0},

      {7.9, 9.5, 4.4, 3.1, 4.4, 2.0, 1.1, 2.4, 1.7, 1.6, 3.9, 7.8},

      {7.2, 6.5, 3.7, 3.1, 4.1, 3.5, 3.7, 2.2, 0.5, 2.2, 5.0, 2.3},
```

```
    {7.2, 7.2, 4.0, 2.9, 3.9, 4.1, 2.3, 0.3, 1.0, 1.7, 1.6, 7.7}
};

float totalRainfall[YEARS] = {0};
float averageYearlyRainfall = 0;
float averageMonthlyRainfall[MONTHS] = {0};

for (int year = 0; year < YEARS; year++) {
    for (int month = 0; month < MONTHS; month++) {
        totalRainfall[year] += rainfall[year][month];
    }
    averageYearlyRainfall += totalRainfall[year];
}
averageYearlyRainfall /= YEARS;

for (int month = 0; month < MONTHS; month++) {
    for (int year = 0; year < YEARS; year++) {
        averageMonthlyRainfall[month] += rainfall[year][month];
    }
    averageMonthlyRainfall[month] /= YEARS;
}



printf("YEAR    RAINFALL (inches)\n");
for (int year = 0; year < YEARS; year++) {
    printf("201%d    %.1f\n", year, totalRainfall[year]);
}

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

printf("\nMONTHLY AVERAGES:\n\n");
```

```
    printf("Jan  Feb  Mar  Apr  May  Jun  Jul  Aug  Sep  Oct  Nov  Dec\n");
    for (int month = 0; month < MONTHS; month++) {
        printf("%.1f  ", averageMonthlyRainfall[month]);
    }
    printf("\n");

    return 0;
}
```

```
PS D:\learning c\output> cd 'd:\learning c\output'
PS D:\learning c\output> & .\'day7_ass14.exe'
YEAR    RAINFALL (inches)
2012    49.8
2013    44.0
2014    43.9

The yearly average is 43.6 inches.

MONTHLY AVERAGES:

Jan  Feb  Mar  Apr  May  Jun  Jul  Aug  Sep  Oct  Nov  Dec
6.8  7.3  4.9  2.9  2.9  2.5  1.8  1.2  1.1  1.7  4.3  6.2
PS D:\learning c\output>
```

CLASS WORK

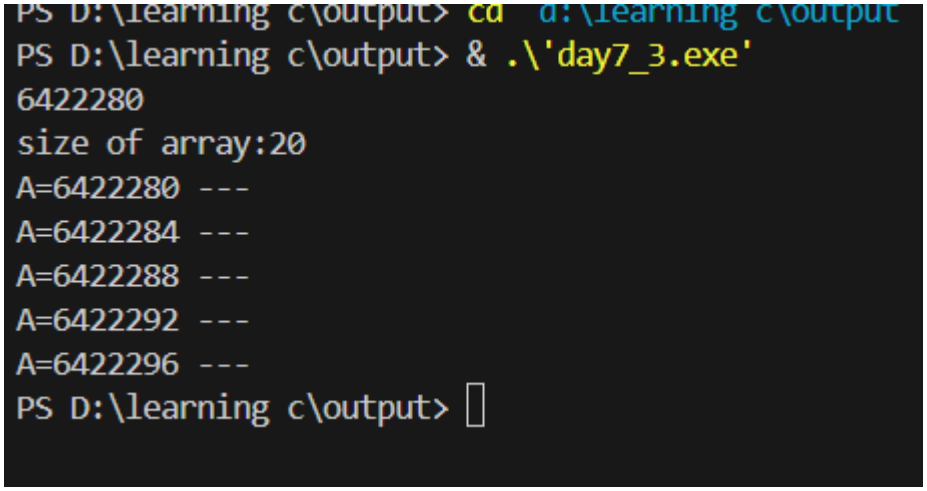
1.
```
#include<stdio.h>
int main()
{
    int const a=50;//local scope
    //const int a=50;
    printf("001 a =%d \n",a);
    int *p;
    p=&a;
```

```
    *p=80;
    printf("002 a=%d \n",a);
}
```

```
PS D:\learning c\output> & .\'day7_1.exe'
001 a =50
002 a=80
PS D:\learning c\output>
```

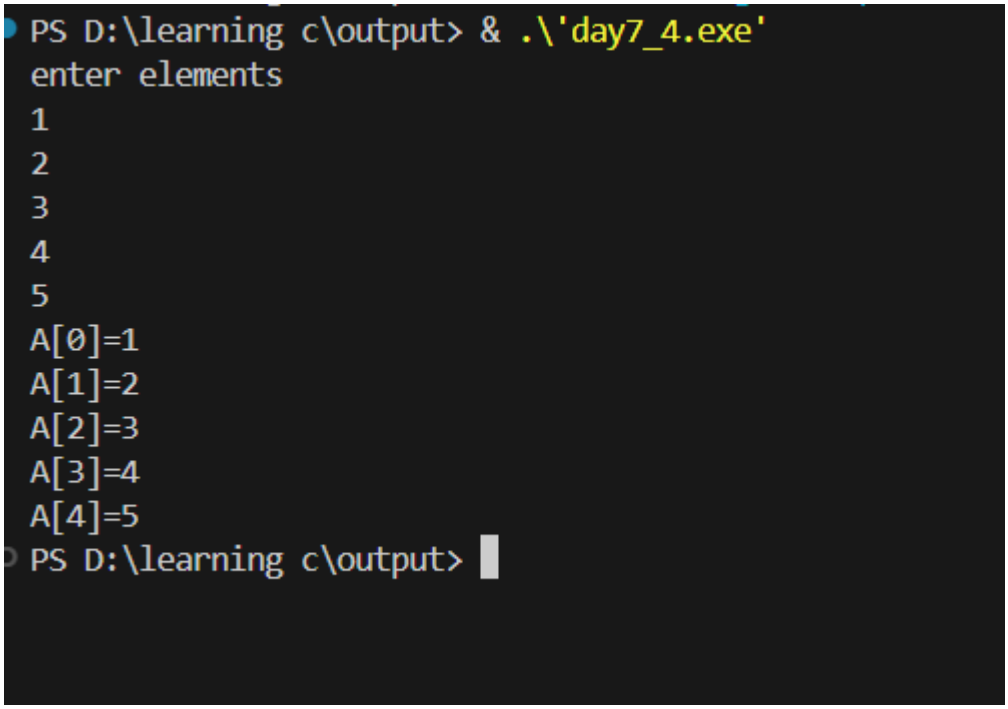
2.
```
#include<stdio.h>
int const a=50;//global scope
int main()
{

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

3.
```
//array
#include<stdio.h>
int main()
{
    int A[5];
    printf("%d \n",A);
    printf("size of array:%d \n",sizeof(A));
    for(int i=0;i<=4;i++)
    {
        printf("A=%d --- \n",(A+i));
    }

}
```

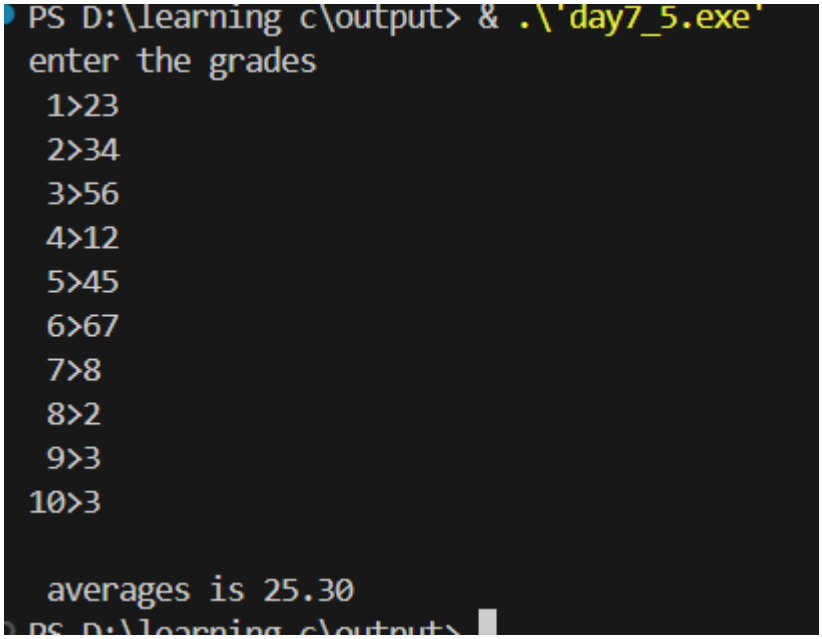


4. #include<stdio.h>

```
int main()
{
   int A[5];
   printf("enter elements \n");
   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]);
   }
}
```



5.
```
#include<stdio.h>
int main()
{
   int grades[10];
   int count=10;
   long sum=0;
   float average=0.0f;
   printf("enter the grades \n");
   for(int i=0;i<count;i++)
   {
      printf("%2u>",i+1);
      scanf("%d",&grades[i]);
      sum+=grades[i];
   }
   average=(float)sum/count;
   printf("\n averages is %.2f \n",average);
   return 0;
}
```



6. #include<stdio.h>

```
int main()
{
   int counters[5]={1,2,3};//only three elements are given here but array size is 5
   for(int i=0;i<5;i++)
   {
      printf("%d \n",counters[i]);//what happens is by default the remaining elements to 0
   }
   printf("array 2 \n");
   int A[3]={[2]=5,[0]=9,[1]=8};//designated intializers
   for(int j=0;j<3;j++)
   {
      printf("%d \n",A[j]);
   }
}
```

```
PS D:\learning c\output> & .\'day7_6.exe'
1
2
3
0
0
array 2
9
8
5
PS D:\learning c\output>
```

7. #include<stdio.h>

#define months 12

int main()

{

int days[months]={31,28,31,30,31,30,31,31,30,31,30,31};

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

{

printf("month %d has %2d days \n",i+1,days[i]);

}

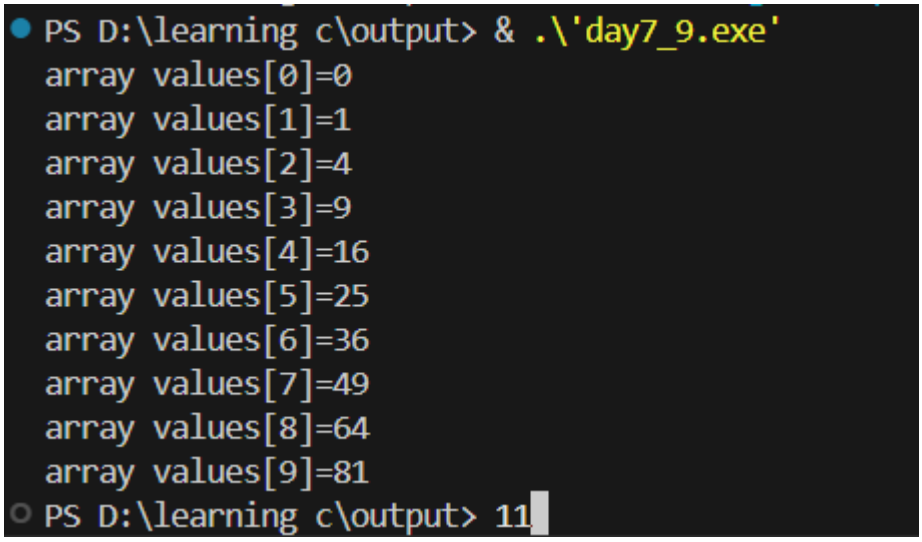
return 0;

}

```
PS D:\learning c\output> & .\'day7_7.exe'
month 1 has 31 days
month 2 has 28 days
month 3 has 31 days
month 4 has 30 days
month 5 has 31 days
month 6 has 30 days
month 7 has 31 days
month 8 has 31 days
month 9 has 30 days
month 10 has 31 days
month 11 has 30 days
month 12 has 31 days
```

9. #include<stdio.h>

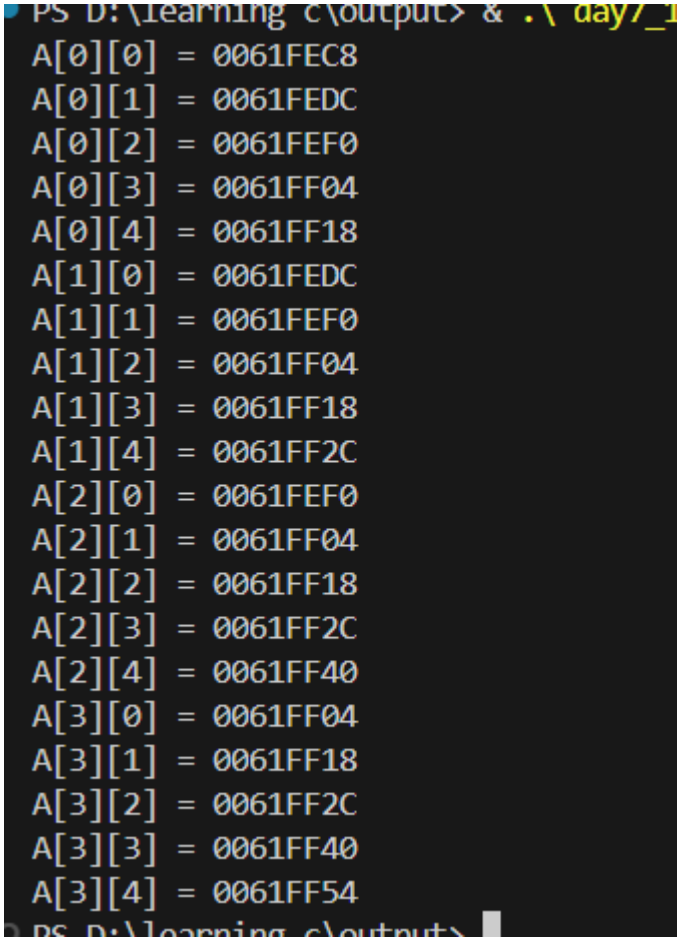
```
int main()
{
   int arrayv[10]={0,1,4,9,16};
   int i;
   for( i=5;i<10;i++)
   {
      arrayv[i]=i*i;
   }
   for(i=0;i<10;i++)
   {
      printf("array values[%i]=%i \n",i,arrayv[i]);
   }
}
```



10. #include <stdio.h>

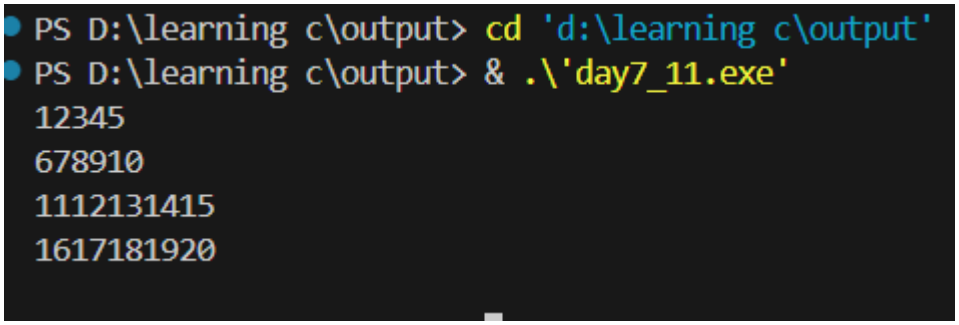
```
int main(){

   int A[4][5] ;

   for(int j = 0; j < 4; j++){
      for(int k = 0; k < 5; k++){
```

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

    return 0;
}
```



11.
```
#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");
    }
    printf("\n");
}
```



12.
```
#include<stdio.h>
int main()
{
    int A[3][3]={[0][0]=1,[1][1]=1,[2][2]=4 };
    for(int j=0;j<3;j++)
    {
        for(int k=0;k<3;k++)
        {
            printf("%d",A[j][k]);
        }
        printf("\n");
    }
    printf("\n");
}
```

```
PS D:\learning c\output> cd 'd:\learning c\output'
PS D:\learning c\output> & .\'day7_12.exe'
100
010
004
```

13. #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 is %d \n ",sum);
}
```

```
PS D:\learning c\output> cd 'd:\learning c\output'
PS D:\learning c\output> & .\'day7_13.exe'
sum is 36
```