

DAY 17

Exercise 1: Write a program to convert English units to metric (i.e., miles to kilometers, gallons to liters, etc.). Include a specification and a code design.

//write a program to convert english units to metric

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

```
void convert_to_km(float);
```

```
void convert_to_litre(float);
```

```
int main()
```

```
{
```

```
    float miles,gallons;
```

```
    printf("enter distance in miles \n");
```

```
    scanf("%f",&miles);
```

```
    //kilometers=miles* 1.60934
```

```
    convert_to_km(miles);
```

```
    printf("enter quantity gallons \n");
```

```
    scanf("%f",&gallons);
```

```
    //Gallons (gal) to Liters (l): Multiply by 3.78541
```

```
    convert_to_litre(gallons);
```

```
}
```

```
void convert_to_km(float miles)
```

```
{
```

```
    float km;
```

```
    km=miles*1.60934;
```

```
    printf("distance in kilometer is %.2f",km);
```

```
}
```

```
void convert_to_litre(float gallons)
```

```
{
```

```
    float litre=gallons*3.78541;
```

```
    printf("gallons to litre:%.2f \n",litre);
```

```
}
```

```
PS D:\learning c\output> cd 'd:\learning c\output'
PS D:\learning c\output> & .\'day17ex1.exe'
enter distance in miles
5
distance in kilometer is 8.05enter quantity gallons
3
gallons to litre:11.36
PS D:\learning c\output> █
```

Exercise 2: Write a program to perform date arithmetic such as how many days there are between 6/6/90 and 4/3/92. Include a specification and a code design.

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

```
#include <stdlib.h>
```

```
#include <stdbool.h> // Include for bool type
```

```
struct date {
```

```
    int day;
```

```
    int month;
```

```
    int year;
```

```
};
```

```
// Function to check if a year is a leap year
```

```
bool isLeapYear(int year) {
```

```
    return (year % 400 == 0) || (year % 100 != 0 && year % 4 == 0);
```

```
}
```

```
// Function to calculate the number of days in a given month
```

```
int daysInMonth(int month, int year) {
```

```
    switch (month) {
```

```
        case 4: case 6: case 9: case 11:
```

```
            return 30;
```

```
        case 2:
```

```

        return isLeapYear(year) ? 29 : 28;
    default:
        return 31;
    }
}

// Function to convert a date to the number of days since 1/1/0000
int date_arithmetic(struct date d) {
    int days = d.day;
    for (int y = 0; y < d.year; y++) {
        days += isLeapYear(y) ? 366 : 365;
    }
    for (int m = 1; m < d.month; m++) {
        days += daysInMonth(m, d.year);
    }
    return days;
}

int main() {
    struct date d1, d2;
    printf("Enter the starting date (dd-mm-yyyy): \n");
    scanf("%d-%d-%d", &d1.day, &d1.month, &d1.year);
    printf("Enter the ending date (dd-mm-yyyy): \n");
    scanf("%d-%d-%d", &d2.day, &d2.month, &d2.year);

    int total_days1 = date_arithmetic(d1);
    int total_days2 = date_arithmetic(d2);
    int difference = abs(total_days1 - total_days2);

    printf("The number of days between the two dates is %d.\n", difference);
    return 0;
}

```

}

```
PS D:\learning c\output> & .\'day17ex2.exe'  
Enter the starting date (dd-mm-yyyy):  
06-06-90  
Enter the ending date (dd-mm-yyyy):  
04-03-92  
The number of days between the two dates is 637.
```

Exercise 3: A serial transmission line can transmit 960 characters each second.
Write a program that will calculate the time required to send a file, given the file's size. Try the program on a 400MB (419,430,400 -byte) file. Use appropriate units. (A 400MB file takes days.)

/*A serial transmission line can transmit 960 characters each second.

Write a program that will calculate the time required to send a file, given the file's size. Try the program on a 400MB (419,430,400 -byte) file. Use appropriate units.

(A 400MB file takes days.)*/

#include <stdio.h>

int main() {

 long long fileSizeInBytes = 419430400; // 400 MB in bytes

 int transmissionRate = 960; // 960 characters per second (1 byte = 1 character)

 // Calculate the total time in seconds

 long long totalTimeInSeconds = fileSizeInBytes / transmissionRate;

 // Convert time into days, hours, minutes, and seconds

 int days = totalTimeInSeconds / (24 * 3600);

 totalTimeInSeconds %= (24 * 3600);

 int hours = totalTimeInSeconds / 3600;

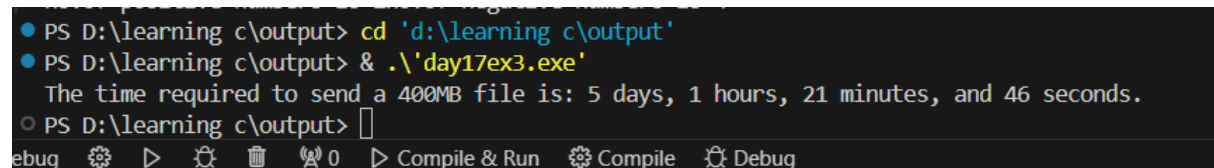
 totalTimeInSeconds %= 3600;

 int minutes = totalTimeInSeconds / 60;

 int seconds = totalTimeInSeconds % 60;

```
printf("The time required to send a 400MB file is: %d days, %d hours, %d minutes, and %d seconds.\n", days, hours, minutes, seconds);
```

```
return 0;\n}\n}
```



```
PS D:\\learning c\\output> cd 'd:\\learning c\\output'\nPS D:\\learning c\\output> & .\\'day17ex3.exe'\n    The time required to send a 400MB file is: 5 days, 1 hours, 21 minutes, and 46 seconds.\nPS D:\\learning c\\output> \nDebug 0  Compile & Run  Compile  Debug
```

Exercise 4: Write a program to add an 8% sales tax to a given amount and round the result to the nearest penny.

//Write a program to add an 8% sales tax to a given amount and round the result to the nearest penny.

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

```
int main()
```

```
{
```

```
    float amount;
```

```
    float tax=0.08;
```

```
    printf("enter an amount \n");
```

```
    scanf("%f",&amount);
```

```
    float amf=amount+(amount*tax);
```

```
    printf("after adding tax to the amount %f \n",amf);
```

```
    int totalamount;
```

```
    totalamount = (int)(amf * 100 + 0.5) / 100.0;
```

```
    printf("result after rounding to nearest penny is %d \n",totalamount);
```

```
}
```

```
/*Manual Rounding Process:
```

Shift Decimal Place: Multiply the total amount by 100 to convert it into an integer with two extra decimal places moved to the right.

1080

×

100

=

108000

Apply Rounding: Add 0.5 to this value to facilitate rounding to the nearest integer.

108000

+

0.5

=

108000.5

Convert to Integer: Use integer conversion to drop the decimal part.

(

i

n

t

)

108000.5

=

108000

Shift Decimal Place Back: Divide by 100 to shift the decimal place back to its original position.

108000

/

100

=

1080.00*/

```

the number of days between the two dates is 637.
PS D:\learning c\output> cd 'd:\learning c\output'
PS D:\learning c\output> & .\'day17-4.exe'
enter an amount
1000
after adding tax to the amount 1080.000000
PS D:\learning c\output> cd 'd:\learning c\output'
PS D:\learning c\output> & .\'day17-4.exe'
enter an amount
1000
after adding tax to the amount 1080.000000
result after rounding to nearest penny is 1080
PS D:\learning c\output> 

```

Exercise 5: Write a program to tell if a number is prime.

//write a program to check if a number is prime

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

```
int main()
```

```
{
```

```
    int n;
```

```
    printf("enter a number \n");
```

```
    scanf("%d",&n);
```

```
    int isprime=1;
```

```
    if(n<=1)
```

```
    {
```

```
        isprime=0;
```

```
    }
```

```
    else
```

```
    {
```

```
        for(int i=2;i<n/2;i++)
```

```
        {
```

```
            //isprime=1;
```

```
            if(n%i==0)
```

```
            {
```

```
                isprime=0;
```

```

        break;
    }

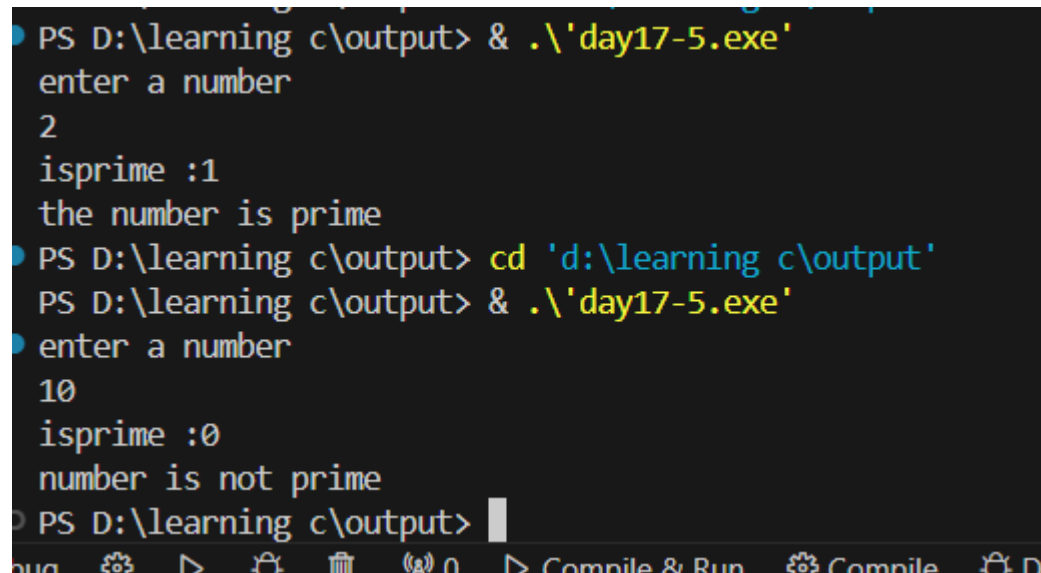
}

}

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

if(isprime)
{
    printf("the number is prime\n");
}
else
{
    printf("number is not prime");
}
}

```



```

PS D:\learning c\output> & .\'day17-5.exe'
enter a number
2
isprime :1
the number is prime
PS D:\learning c\output> cd 'd:\learning c\output'
PS D:\learning c\output> & .\'day17-5.exe'
enter a number
10
isprime :0
number is not prime
PS D:\learning c\output>

```

Exercise 6: Write a program that takes a series of numbers and counts the number of positive and negative values.

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

```
int main()
```

```
{
```

```
    int n;
```



```
printf("enter the limit\n");
scanf("%d",&n);
printf("enter the numbers\n");
int num[n];
for(int i=0;i<n;i++)
{
    scanf("%d",&num[i]);
}
int pcount=0;
int ncount=1;
for(int i=0;i<n;i++)
{
    if(num[i]>0)
    {
        printf("%d:positive \n",num[i]);
        pcount++;
    }
    else
    {
        printf("%d:not positive \n",num[i]);
        ncount++;
    }
}
printf("no.of positive numbers is \n %d",pcount);
printf("no.of negative numbers is \n %d",ncount);
}
```

```

PS D:\learning c\output> & .\'day17ex-6.exe'
enter the limit
5
enter the numbers
1
0
-1
-2
3
1-positive
0-not positive
-1-not positive
-2-not positive
3-positive
no.of positive numbers is 2no.of negative numbers is 4
PS D:\learning c\output> █

```

1. C program to find the HCF (Highest Common Factor) of given numbers using recursion
2. C program to find the LCM (Lowest Common Multiple) of given numbers using recursion
3. C program to find the GCD (Greatest Common Divisor) of given numbers using recursion
4. C program to convert a Decimal number to Binary using Recursion.
5. C program to convert a Binary number to Gray Code
6. C program to convert a Binary number to Gray Code using Recursion
7. C program to print following Pyramid:

```

*****
****  ****
***   ***
**    **
*     *

```
8. C program to find the sum of Natural Number/Factorial of Number of all natural numbers from 1 to N.
Series: $1/1! + 2/2! + 3/3! + 4/4! + \dots N/N!$
9. C program to find sum of following series:
 $1 + 3^2/3^3 + 5^2/5^3 + 7^2/7^3 + \dots$ till N terms
10. C program to replace all EVEN elements by 0 and Odd by 1 In One Dimensional Array
11. C Program to Read a Matrix and Print Diagonals
12. C program to print the upper triangular portion of a 3x3matrix
13. C program to input and print text using Dynamic Memory Allocation.
14. C program to read a one dimensional array, print sum of all elements along with inputted array elements using Dynamic Memory Allocation.

1.//finding hcf of a given numbers using recursion

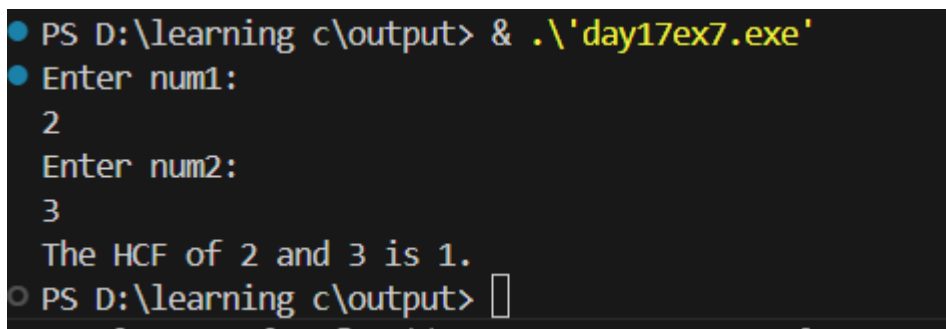
```
#include <stdio.h>
```

```
// Function to find the HCF using recursion
int findHCF(int a, int b) {
    if (b == 0)
        return a;
    else
        return findHCF(b, a % b);
}

int main() {
    int num1, num2;
    printf("Enter num1: \n");
    scanf("%d", &num1);
    printf("Enter num2: \n");
    scanf("%d", &num2);

    int hcf = findHCF(num1, num2);
    printf("The HCF of %d and %d is %d.\n", num1, num2, hcf);

    return 0;
}
```



```
PS D:\learning c\output> & .\'day17ex7.exe\'
Enter num1:
2
Enter num2:
3
The HCF of 2 and 3 is 1.
PS D:\learning c\output>
```

2. #include<stdio.h>

```
int findLcm(int,int);
```

```
int findGcd(int,int);
```

```
int main()
```

```
{  
    int num1,num2;  
    printf("enter the first number \n");  
    scanf("%d",&num1);  
    printf("enter second number \n");  
    scanf("%d",&num2);  
    /*lcm=num1*num2/gcd(num1,num2)*/  
    int lcm=findLcm(num1,num2);  
    printf("lcm is %d",lcm);  
}
```

```
int findGcd(int num1,int num2)
```

```
{  
    if(num2==0)  
    {  
        return num1;  
    }  
    else  
    {  
        return findGcd(num2,num1%num2);  
    }  
}
```

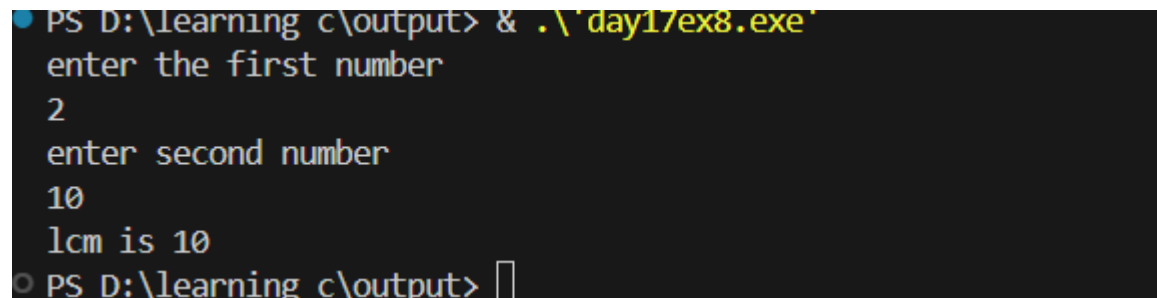
```
int findLcm(int num1,int num2)
```

```
{  
    int l=num1*num2;  
    if(num2==0)  
    {  
        return num1;  
    }  
    else  
    {
```

```

        return l/findGcd(num2,num1%num2);
    }
}

```



```

PS D:\learning c\output> & .\day17\ex8.exe
enter the first number
2
enter second number
10
lcm is 10
PS D:\learning c\output>

```

3. //finding gcd using recursion

```

#include<stdio.h>

int findGcd(int,int);

int main()
{
    int num1,num2;
    printf("enter the first number \n");
    scanf("%d",&num1);
    printf("enter second number\n");
    scanf("%d",&num2);
    int gcd=findGcd(num1,num2);
    printf("the gcd is %d \n",gcd);

}

int findGcd(int num1,int num2)
{
    if(num2==0)
    {
        return num1;
    }
    else
    {
        return findGcd(num2,num1%num2);
    }
}

```

```
}  
}
```

```
PS D:\learning c\output> cd 'd:\learning c\output'  
PS D:\learning c\output> & .\'day17ex9.exe'  
enter the first number  
2  
enter second number  
10  
the gcd is 2  
PS D:\learning c\output> 
```

4. //convert decimal to binary

```
#include<stdio.h>  
  
int deci_binary(int);  
  
int main()  
{  
    int deci;  
  
    printf("enter a decimal number \n");  
  
    scanf("%d",&deci);  
  
    deci_binary(deci);  
}  
  
int deci_binary(int deci)  
{  
    if(deci>1)  
    {  
        deci_binary(deci/2);  
    }  
  
    int remainder=deci%2;  
    printf("%d",remainder);  
}
```

```

● PS D:\learning c\output> cd 'd:\learning c\output'
PS D:\learning c\output> & .\day17ex10.exe
● enter a decimal number
10
1010
○ PS D:\learning c\output> █

```

5. //binary number to gray code

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

```
int main()
```

```
{
```

```
    int binary[4];
```

```
    printf("enter a binary number \n");
```

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

```
    {
```

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

```
    }
```

```
    int n=4;
```

```
    int gray[n];
```

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

```
    {
```

```
        if(i==0)
```

```
        {
```

```
            gray[i]=binary[i];
```

```
        }
```

```
        else
```

```
        {
```

```
            gray[i]=binary[i] ^ binary[i-1];
```

```
        }
```

```
    }
```

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

```
    {
```

```
        printf("%d",gray[i]);
```

```
}  
}
```

```
● PS D:\learning c\output> & .\'day17ex11.exe'  
enter a binary number  
1  
0  
1  
1  
1110  
○ PS D:\learning c\output> █
```

6. //binary to decimal using recursion

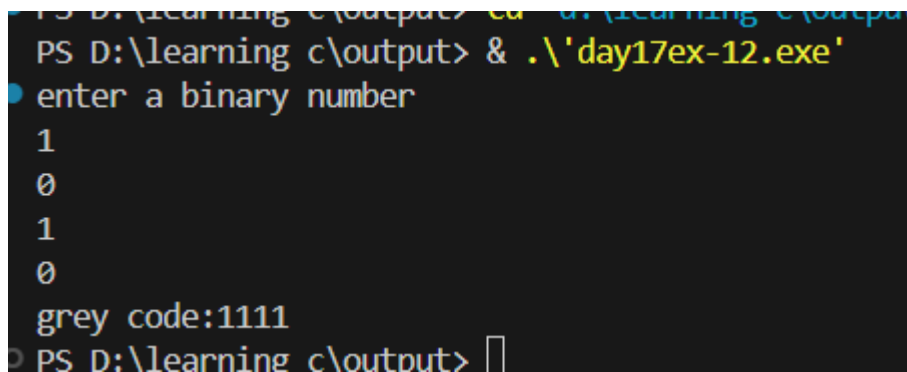
```
#include<stdio.h>  
  
void binarytogrey(int[],int[],int,int);  
  
int main()  
{  
    int binary[4];  
    printf("enter a binary number \n");  
    for(int i=0;i<4;i++)  
    {  
        scanf("%d",&binary[i]);  
    }  
    int n=4;  
    int gray[n];  
    binarytogrey(binary,gray,n,0);  
    printf("grey code:");  
    for(int i=0;i<n;i++)  
    {  
        printf("%d",gray[i]);  
    }  
}  
  
void binarytogrey(int binary[],int gray[],int n,int index)  
{
```



```

if(index==n)
{
    return ; //cz end of the array
}
if(index==0)
{
    gray[index]=binary[index];
}
else
{
    gray[index]=binary[index] ^ binary[index-1];
}
binarytogrey(binary,gray,n,index+1);
}

```



```

PS D:\learning c\output> & .\'day17ex-12.exe\'
enter a binary number
1
0
1
0
grey code:1111
PS D:\learning c\output>

```

7. #include <stdio.h>

```

int main() {

    int rows = 5; // Number of rows in the pyramid

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

        // Print leading spaces

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

            printf(" ");

        }
    }
}

```

```

// Print stars for the left half
for (int j = 0; j < rows - i; j++) {
    printf("*");
}

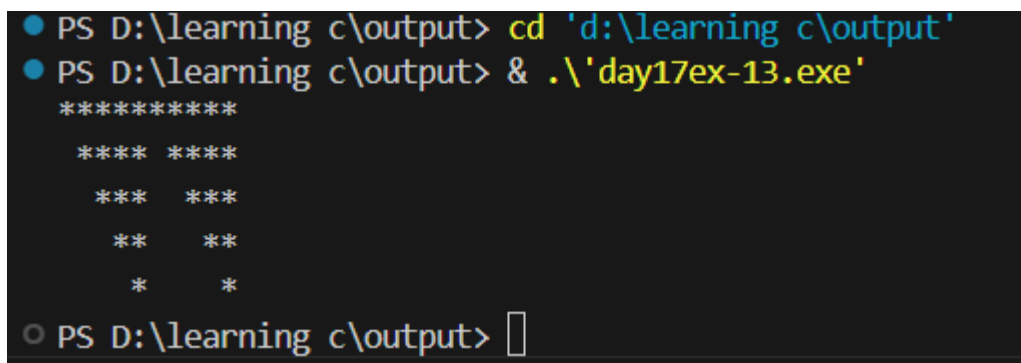
// Print spaces in the middle
for (int j = 0; j < i; j++) {
    printf(" ");
}

// Print stars for the right half
for (int j = 0; j < rows - i; j++) {
    printf("*");
}

printf("\n");
}

return 0;
}

```



```

PS D:\learning c\output> cd 'd:\learning c\output'
PS D:\learning c\output> & .\'day17ex-13.exe'
*****
*****
***   ***
**    **
*     *
*     *
*****
PS D:\learning c\output>

```

8. #include <stdio.h>

```

int factorial(int n) {
    if (n == 0 || n == 1) {
        return 1;
    }
    return n * factorial(n - 1);
}

```

```
int main() {  
  
    int n;  
  
    printf("Enter limit:\n");  
    scanf("%d", &n);  
  
    float natural[n];  
  
    printf("Enter the natural numbers:\n");  
    for (int i = 0; i < n; i++) {  
        scanf("%f", &natural[i]);  
    }  
  
    float fact[n];  
  
    for (int i = 0; i < n; i++) {  
        fact[i] = factorial(natural[i]);  
    }  
  
    float dig[n];  
  
    for (int i = 0; i < n; i++) {  
        dig[i] = (float)natural[i] / fact[i];  
    }  
  
    float sum = 0.0;  
  
    for (int i = 0; i < n; i++) {  
        sum += dig[i];  
    }  
  
    printf("The sum is %f\n", sum);  
  
    return 0;  
}
```

```
Enter limit:
5
Enter the natural numbers:
1
2
3
4
5
The sum is 2.708333
PS D:\learning c\output> 
```

9. #include<stdio.h>

int main()

{

int n;

printf("enter the limit \n");

scanf("%d",&n);

float sum=0.0;

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

{

if(i%2!=0)

{

sum+=(float)((i*i)/(i*i*i));

}

}

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

}

```
PS D:\learning c> cd 'd:\learning c\output'
PS D:\learning c\output> & .\'day17ex-15.exe'
enter the limit
10
sum is 1.000000
PS D:\learning c\output> █
```

10. #include<stdio.h>

int main()

{

int arr[10];

printf("enter the elements to the array \n");

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

{

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

}

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

{

if(arr[i]%2==0)

{

arr[i]=0;

}

else if(arr[i]%2!=0)

{

arr[i]=1;

}

}

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

{

printf("%d",arr[i]);

```
        printf("\n");  
    }  
}
```

PS D:\learning c\output> & .\day17ex-16.exe'

enter the elements to the array

2

4

6

8

1

3

5

7

9

1

0

0

0

0

1

1

1

1

1

1

PS D:\learning c\output>

11. //read a matrix and print the diagonals

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

```
int main()
```

```
{
```

```
    int rows,column;
```

```
printf("enter the no.of rows \n");
scanf("%d",&rows);
printf("enter the no.of columns \n");
scanf("%d",&column);
int m[rows][column];
printf("enter the elements \n");
for(int i=0;i<rows;i++)
{
    for(int j=0;j<column;j++)
    {
        scanf("%d",&m[i][j]);
    }
}
printf("diagonal elements \n");
for(int i=0;i<rows;i++)
{
    for(int j=0;j<column;j++)
    {
        if(i==j)
        {
            printf("%d",m[i][j]);
        }
    }
}
}
```

```
PS D:\learning c\output> cd ..\learning c\output
PS D:\learning c\output> & .\'day17ex-18.exe'
enter the no.of rows
2
enter the no.of columns
2
enter the elements
1
2
3
4
diagonal elements
14
PS D:\learning c\output> █
```

12. //print upper triangular port of 3x3 matrix

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

```
int main()
```

```
{
```

```
    int m[3][3];
```

```
    printf("enter the elements to the matrix \n");
```

```
    for(int i=0;i<3;i++)
```

```
    {
```

```
        for(int j=0;j<3;j++)
```

```
        {
```

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

```
        }
```

```
    }
```

```
    for(int i=0;i<3;i++)
```

```
    {
```

```
        for(int j=0;j<3;j++)
```

```
        {
```

```
            if((i==j) || i<j)
```

```
            {
```

```
                printf("%d",m[i][j]);
```



```

    }
}
}
}
}

```

```

enter the elements to the matrix
1
2
3
4
5
6
7
8
9
123569

```

13. #include <stdio.h>

#include <stdlib.h>

```
int main() {
```

```
    char *text;
```

```
    int size;
```

```
    printf("Enter the size of the text: ");
```

```
    scanf("%d", &size);
```

```
    // Allocate memory dynamically
```

```
    text = (char *)malloc((size + 1) * sizeof(char)); // +1 for the null terminator
```

```
    if (text == NULL) {
```

```
        printf("Memory allocation failed!\n");
```

```
        return 1; // Exit if memory allocation fails
    }
```

```

}

printf("Enter the text: ");

scanf(" ");

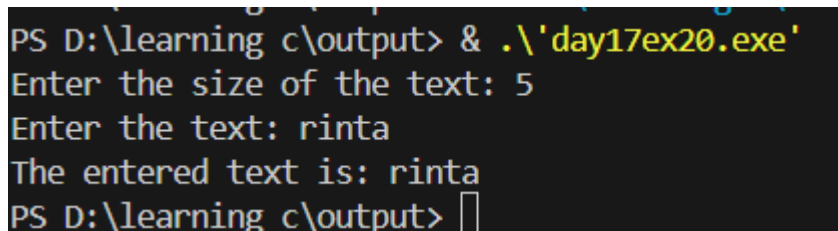
scanf("%[^\n]", text); // Read the text from user until newline

printf("The entered text is: %s\n", text);

// Free allocated memory
free(text);

return 0;
}

```



```

PS D:\learning c\output> & .\'day17ex20.exe\'
Enter the size of the text: 5
Enter the text: rinta
The entered text is: rinta
PS D:\learning c\output>

```

14. #include <stdio.h>

#include <stdlib.h>

```

int main() {
    int n;
    int *array;
    int sum = 0;

    printf("Enter the number of elements: ");
    scanf("%d", &n);

    // Dynamically allocate memory for the array
    array = (int *)malloc(n * sizeof(int));

```

```
if (array == NULL) {  
    printf("Memory allocation failed!\n");  
    return 1; // Exit if memory allocation fails  
}  
  
printf("Enter the elements of the array:\n");  
for (int i = 0; i < n; i++) {  
    scanf("%d", &array[i]);  
    sum += array[i]; // Calculate sum while reading input  
}  
  
printf("The elements of the array are:\n");  
for (int i = 0; i < n; i++) {  
    printf("%d ", array[i]);  
}  
printf("\n");  
  
printf("The sum of all elements is: %d\n", sum);  
  
// Free allocated memory  
free(array);  
  
return 0;  
}
```

```
PS D:\learning c\output> & .\ day17e
Enter the number of elements: 3
Enter the elements of the array:
1
2
3
The elements of the array are:
1 2 3
The sum of all elements is: 6
PS D:\learning c\output> 
```