PHASE 1 DAY17

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.

answer:

```
#include<stdio.h>
int main()
{
    int choice;
    int miles,gallons;
    printf("Enter the type of conversion :\n1.Miles to kilometers\n2.Gallons to liters\nEnter your choice:");
    scanf("%d",&choice);
    switch(choice)
    {
        case 1:
        {
          printf("\nEnter Miles:");
          scanf("%d",&miles);
          printf("%d miles is %f Kilometers",miles,(float)(miles*1.60));
        break;
        }
        case 2:
       {
          printf("\nEnter Gallons:");
          scanf("%d",&gallons);

          printf("\n%d gallons is %f liters",gallons,(float)(gallons*3.78));

        break;

        }

        default:

        printf("Enter from menu");

    }

    return 0;

    }
```

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>



int is_leap_year(int year)

{

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

}



int days_in_month(int month, int year)

{

    switch (month)

    {

        case 1: case 3: case 5: case 7: case 8: case 10: case 12:

            return 31;
        case 4: case 6: case 9: case 11:
            return 30;

        case 2:
            return is_leap_year(year) ? 29 : 28;
        default:
            return 0;
    }
}

int date_to_days(int day, int month, int year)
{
    int total_days = 0;

    for (int y = 1; y < year; y++)
    {
        total_days += is_leap_year(y) ? 366 : 365;
    }

    for (int m = 1; m < month; m++)
    {
        total_days += days_in_month(m, year);
    }
    total_days += day;


    return total_days;

}

int main()

{

    int day1, month1, year1;

    int day2, month2, year2;


    printf("Enter the first date (dd/mm/yyyy): ");

    ("%d/%d/%d", &day1, &month1, &year1);


    printf("Enter the second date (dd/mm/yyyy): ");

    ("%d/%d/%d", &day2, &month2, &year2);


    int days1 = date_to_days(day1, month1, year1);

    int days2 = date_to_days(day2, month2, year2);

    int difference = days2 - days1;

    if (difference < 0)

    {

        difference = -difference;

    }

    printf("The number of days between %d/%d/%d and %d/%d/%d is %d days.\n",day1, month1,
year1, day2, month2, year2, difference);
```

```
    return 0;
}
```

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 prog ram on a 400MB (419,430,400 -byte) file. Use appropriate units. (A 400MB file takes days.)

Answer:

```
#include<stdio.h>
int main()
{
    int size;
    printf("Enter the size of file in MBs:");
    scanf("%d",&size);
    // no of seconds a day = 24*60*60= 86400sec
    //400 MB ->86400 : 216 sec for 1 MB
  float time_taken=size*216;
  float hrs=time_taken/(3600);
  printf("Time taken=%.1f",hrs);
    return 0;
}
```

Exercise 4: 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()
{
    double amount, tax_rate = 0.08, tot_amount;
    int round_total;
    printf("Enter the original amount: ");
    scanf("%lf",&amount);
    tot_amount = amount + (amount * tax_rate);
   // round to the nearest penny-> Multiply by 100 add 0.5 and then divide by 100
        round_total = (int)(tot_amount * 100 + 0.5);
        tot_amount = (double)round_total / 100;
    printf("The total amount after adding 8%% sales tax is: %.2f\n", tot_amount);
    return 0;
}
```

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

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

    if(isprime(num))
    {
        printf("Number is prime!");
    }
    else

    {

        printf("Number is not prime!");

    }

    return 0;

}

int isprime(int n)

{

    int flag=1;

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

    {

        if(n%i==0)

        {

            flag=0;

        }

    }

    return flag;

}
```

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 size,ncount=0,pcount=0;
    printf("Enter the size:");
    scanf("%d",&size);
    int arr[size];

    printf("Enter %d numbers:",size);
    for(int i=0;i<size;i++)
    {
        scanf("%d",&arr[i]);

        if(arr[i]>0)
        {
            pcount++;
        }
        else
        {
            ncount++;
        }
    }
    printf("\nYou have entered %d positive and %d negative numbers.",pcount,ncount);
    return 0;

}
```

Print pattern

```
**********
****  ****
***    ***
**      **
*        *
```

```
#include <stdio.h>

int main()
 {
    for (int rows = 1; rows <= 5; rows++)
    {
        for (int cols = 1; cols <= 10; cols++)
        {
            if ((cols == 1) || (cols == 10) || (rows+cols<=6)||((cols-rows)>=5)&&(cols>rows)
&&(rows+cols>=9)|| rows==1)
            {
                printf("*");
            }
            else
            {
                printf(" ");
            }
        }
        printf("\n");
    }

    return 0;
}
```

2. C program to replace all even nos by 0 and odd nos by 1 in an array

```
#include <stdio.h>

int main()

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

    printf("\nArray :");

    for(int j=0;j<n;j++)
    {
        printf("%d ",arr[j]);
    }

    for(int k=0;k<n;k++)
    {
        if(0==arr[k]%2)
        {
            arr[k]=0;
        }
        else
        {
            arr[k]=1;
        }
    }
    printf("\nArray after replacing :");

    for(int j=0;j<n;j++)
    {
        printf("%d ",arr[j]);
    }
    return 0;
}
```

3.program to find HCF GDC & LCM

```
#include <stdio.h>

int main()
{
    int num1, num2;

    // Input two numbers
    printf("Enter two numbers: ");
    scanf("%d %d", &num1, &num2);

    int gcd_result = gcd(num1, num2);
    int lcm_result = lcm(num1, num2);
    int hcf_result = hcf(num1, num2);

    printf("GCD of %d and %d is: %d\n", num1, num2, gcd_result);
    printf("LCM of %d and %d is: %d\n", num1, num2, lcm_result);
    printf("HCF of %d and %d is: %d\n", num1, num2, hcf_result);

    return 0;
}

int gcd(int a, int b)
 {
    if (b == 0)
 {
        return a;
    }
    return gcd(b, a % b);
}
```

```
int lcm(int a, int b)
 {
    return (a * b) / gcd(a, b);
}


int hcf(int a, int b)
 {
    return gcd(a, b);
}
```

4.Read a matrix and print diagonal elements

```
#include <stdio.h>

int main()

{
    int rows,cols;
    printf("Enter the size of matrix (rows*cols):");
    scanf("%d %d",&rows,&cols);
 int arr[rows][cols];
 printf("\nEnter elements");
    for(int i=0;i<rows;i++)
    {
        for(int j=0;j<cols;j++)
        {
            scanf("%d",&arr[i][j]);
        }
        printf("\n");
    }
    for(int i=0;i<rows;i++)
    {
        for(int j=0;j<cols;j++)
        {
            printf("%d ",arr[i][j]);
        }
        printf("\n");
    }
    printf("The diagonal elements are:\n");
    for(int i=0;i<rows;i++)
    {
        for(int j=0;j<cols;j++)
        {
            if(i==j)
            {
                printf("(%d,%d)->%d\n",i,j,arr[i][j]);
            }
        }
    }

    return 0;
}
```

Find sum of natural/factorial of number

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

{
    int num;
    int sum=0;
    printf("Enter number:");
    scanf("%d",&num);
    for(int i=1;i<num;i++)
    {
      sum += (i/fact(i));
    }
    printf("Sum = %d",sum);
    return 0;
}

int fact(int n)
{
    if(n==0||n==1)
    {
    return 1;
    }
    else
    {
    return n*fact(n-1);
    }
}
```

Sum of following series:
1+3^2/3^3+5^2/5^3+7^2+7^3….N

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

{
    int num;
    float sum=0;
    printf("Enter number:");
    scanf("%d",&num);

    for(int i=1;i<num;i+=2)
    {
      sum += (float)(square(i))/(cube(i));
    }
    printf("Sum = %.2f",sum);

    return 0;
}

int square(int n)
{
    return n*n;
}

int cube(int n)
{
    return n*n*n;
}
```

program to convert decimal to binary

```
#include <stdio.h>
void decimalToBinary(int n)
{
 if (n == 0)
 {
 return;
 }
 decimalToBinary(n / 2);
 printf("%d", n % 2);
}
int main()
{
 int num;
 printf("Enter a decimal number: ");
 scanf("%d", &num);
 printf("Binary representation of %d is: ", num);
 if (num == 0)
 {
 printf("0");
 }
 else
 {
 decimalToBinary(num);
 }
 printf("\n");
 return 0;
}
```

binary to gray

```
#include <stdio.h>

int binaryToGray(int n)
{
    return n ^ (n >> 1);
}

void printGrayCode(int n)
{
    int gray = binaryToGray(n);
    int bits = sizeof(n) * 8;

    int flag= 1;

    for (int i = bits - 1; i >= 0; i--)
    {
        int bit = (gray >> i) & 1;
        if (bit == 1)
        {
            flag = 0;
        }
        if (!flag)
        {
            printf("%d", bit);
        }
    }
}

int main()
{
    int num;
    printf("Enter a decimal number: ");
    scanf("%d", &num);

    printf("Gray code of %d is: ", num);

    if (num == 0)
    {
        printf("0");
    } else {
        printGrayCode(num);
    }

    printf("\n");
    return 0;
}
```

C program to print the upper triangular portion of a matrix

```
#include <stdio.h>
int main() {
 int matrix[3][3], i, j;
 printf("Enter 3x3 matrix elements:\n");
 for (i = 0; i < 3; i++) {
 for (j = 0; j < 3; j++) {
 scanf("%d", &matrix[i][j]);
 }
 }
 printf("Upper triangular portion:\n");
 for (i = 0; i < 3; i++) {
     for (j = 0; j < 3; j++) {
 if (j >= i)
 printf("%d ", matrix[i][j]);
 else
 printf(" "); // Empty space for formatting
 }
 printf("\n");
 }
 return 0;
}
```

C program to input and print text using Dynamic Memory Allocation.

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

int main()
{
    char *text;
    int n;
    printf("Enter the number of characters: ");
    scanf("%d", &n);
    text = (char *)malloc((n + 1) * sizeof(char));
    if (text == NULL) {
        printf("Memory allocation failed.\n");
        return 1;
    }
    printf("Enter text: ");
    getchar();
    for (int i = 0; i < n; i++) {
        text[i] = getchar();
        if (text[i] == '\n') {
            break;
        }
    }
    text[n] = '\0';
    printf("You entered: %s\n", text);
    free(text);
    return 0;
}
```

18. C program to read a one dimensional array, print sum of all elements along with inputted array elements using Dynamic Memory Allocation.

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

int main()
{
    char *buffer;
    int length;
    printf("Enter the number of characters: ");
    scanf("%d", &length);
    buffer = (char *)malloc((length + 1) * sizeof(char));
    if (buffer == NULL) {
        printf("Memory allocation failed.\n");
        return 1;
    }
    printf("Enter text: ");
    getchar();
    for (int i = 0; i < length; i++) {
        buffer[i] = getchar();
        if (buffer[i] == '\n') {
            break;
        }
    }
    buffer[length] = '\0';
    printf("You entered: %s\n", buffer);
    free(buffer);
    return 0;
}
```