1. All years that are divisible by 400 or by 4 and not by 100 are leap years. Write a C program which accepts a year as input and displays an appropriate message which informs the user whether or not the year is a leap year.

```
void Judge_Leap_Year(int year){
    if(year%400==0){
        printf("%d year is Leap Year!\n",year);
    }
    else if(year%4==0 && year%100!=0){
        printf("%d year is Leap Year!\n",year);
    }
    else{
        printf("%d year is not Leap Year!\n",year);
    }
}
```

2. Design, write and test a program which reads in a list of examination marks (one per line) until a mark greater than 100 is entered. Your program should then print out the total of the marks entered and the average (arithmetic mean) mark. When that is working, convert it to calculate the geometric mean.

```
void Calculate_ALL_MEANS(void){

    int total=0;
    int times=1;
    int num=0;
    int temp=0;
    setvbuf(stdout,NULL,_IONBF,0);
    for(int i = 0;;i++){
    printf("     Student Mark :     ");
    scanf("%d",&temp);
    printf("\n");
    total+=temp;
    times*=temp;
    num++;
    if(temp>100){
        break;
    }
    }

    printf(" Total Marks :     %d     \n Arithmetic Mean :     %d     \n Geometric
Mean :      %d      \n",total,total/num,(int) pow((double) times,1.0/num));

}
```

3. Making use of a switch statement (based on the month) to write a program to read in a month and day. Your program should then print out an appropriate error message ONLY when an invalid date is input. (Ignore leap years!)

```
void Judge_Month_and_Day(void){
    int month = 0;
    int day = 0;
    printf("     Month :      ");
    setvbuf(stdout,NULL,_IONBF,0);
    scanf("%d",&month);
    printf("\n     Day :      ");
    scanf("%d",&day);
    printf("\n");

    switch(month){
    case 1 :{
        if(day<=31 && day>0) return;
        break;
    };

    case 2 :{
        if(day<=28 && day>0) return;
        break;
    };

    case 3 :{
        if(day<=31 && day>0) return;
        break;
    };

    case 4 :{
        if(day<=30 && day>0) return;
        break;
    };

    case 5 :{
```

```
    if(day<=31 && day>0) return;
    break;
};

case 6 :{
    if(day<=30 && day>0) return;
    break;
};

case 7 :{
    if(day<=31 && day>0) return;
    break;
};

case 8 :{
    if(day<=31 && day>0) return;
    break;
};

case 9 :{
    if(day<=30 && day>0) return;
    break;
};

case 10 :{
    if(day<=31 && day>0) return;
    break;
};

case 11 :{
    if(day<=30 && day>0) return;
    break;
};

case 12 :{
    if(day<=31 && day>0) return;
    break;
};

default : break;
}


printf("Error in data - don’t understand Month : %d,
Day :      %d\n",month,day);

}
```

4.

```
void Bowling(void){

    int total=0;
    int bowl[5]={0,0,0,0,0};
    int temp_Score=0;
    setvbuf(stdout,NULL,_IONBF,0);
    for(int i = 0; i<5 ; i++){
        for(int j = 0; j<3 ; j++){
            printf(" %d%s Bowler, Game %d    \n Score :      ",i+1,i>2 ? "th" : i==0 ?
"st" : i==1 ? "nd" : "rd",j+1);
            scanf("%d",&temp_Score);
            printf("\n");
            bowl[i]+=temp_Score;
            total+=temp_Score;
        }
    }

    for(int k = 0; k<5 ; k++){
        printf(" %d%s Bowler's Average Game Score :      %d\n",k+1,k>2 ? "th" :
k==0 ? "st" : k==1 ? "nd" : "rd",bowl[k]/3);
    }
    printf(" Average Bowlers' Game Score :      %d\n",total/15);
}
```

4. Write a program which declares an array named slopes, with its elements initialised (in the declaration statement) to the following list of values: 17.24, 25.63, 5.94, 33.92, 3.71, 32.84, 35.93, 18.24, 6.92. Your program should search through the array, to find both the maximum and minimum values in the array, and then print those values. Remember to use flowchart or pseudocode to plan the algorithm for your program, along with the variables needed, in your daybook, before trying to write any C language code

```
void Find_Maximum_and_Minimum(void){
    double slope[9] = {17.24, 25.63, 5.94, 33.92, 3.71,
            32.84, 35.93, 18.24, 6.92};

    int maximum = slope[0];
    int minimum = slope[0];

    for(int i = 0; i < 9; i++){
        if (slope[i] > maximum){
            maximum=slope[i];
        }
        else if(slope[i] < minimum){
            minimum=slope[i];
        }
    }

    printf(" Maximum :      %d \n Minimum :      %d\n",maximum,minimum);

}
```