Name:Ramil Ashraf
RegNo.:21BCE3454
Class code: BCSE101E
Slot: TG1_L45+L46+L55+L56

Q1. Write a problem analysis chart (PAC), flowchart and Algorithm to calculate the age of a housefly in seconds, given the number of days the housefly lived. [CO1] [L1] For example, if a housefly lived for 21 days, then its approximate age in seconds is 21*24*60*60 is 1814400.

Answer:

| Data	Process	Output
1. No of days Housefly has lived	1. Input days 2. Age = days*24*60*60	1. Print age

Flowchart:
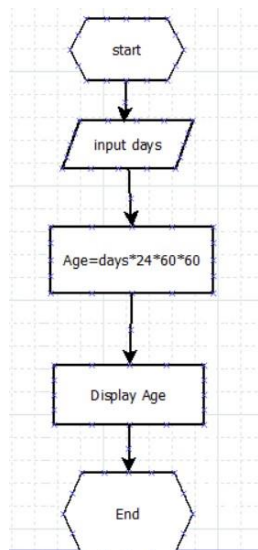
Algorithm:

1.Read No. of days

2. Age = Days*24*60*60

3. Display Age of Housefly

Q2. Milk is collected for sales from nearest 'n' farms to the milk booth. Given the amount of milk from 'n' farms in liters and ml. Write a PAC chart, flowchart, and algorithm to compute total quantity of milk in the booth.

Answer:

| Data	Process	Output
1. n farms	1. read l1,m1,l2,m2 2. Add them 3.divide by 1000 4. quotient is liters remainder part is ml	1. display quantity in liters and milliliters

Flowchart:
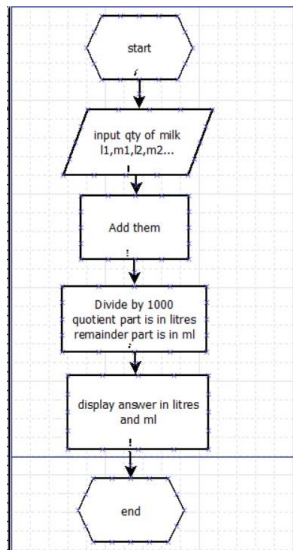
Algorithm:

1. input n farms
2. Input qty of milk in l1,m1,l2,m2...
3. Add them
4. Divide by 1000, quotient part is in liter remainder part is ml
5. Display answer

Q3. Write a PAC Chart, flowchart and algorithm for converting the given two-digit number into its corresponding Roman numeral

Answer:

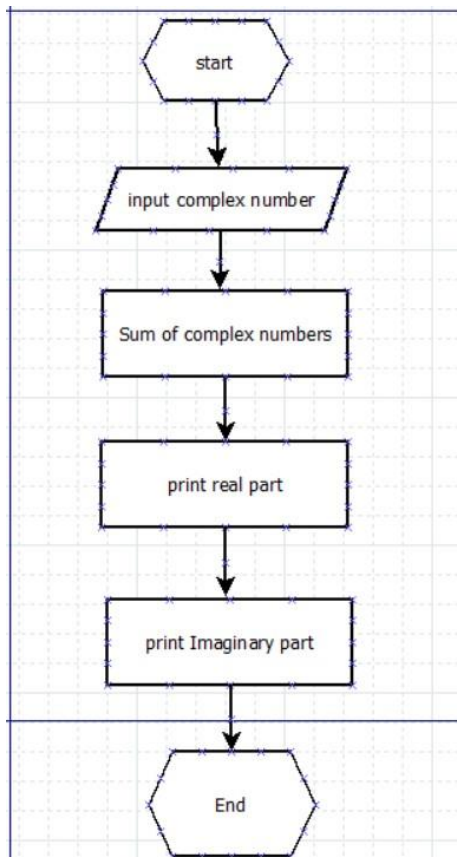| Data	Process	Output
1. 2-digit Number	1. Input the 2-digit number 2. Declare the roman numerals of numbers such as 1,4,9,10 etc. 3. Subtract the largest no. that is less than or equal to the given no.	1. print Roman number of given 2 digit number

Flowchart:
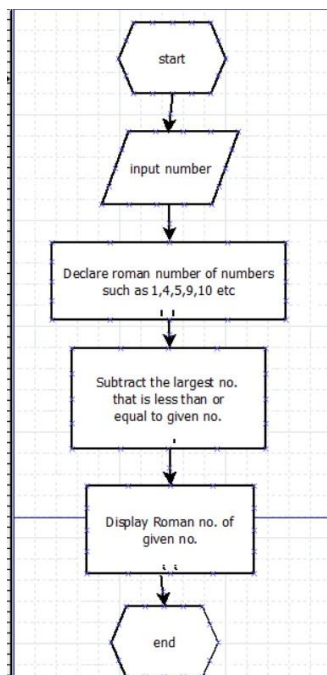
Algorithm:

1. input number

2. Declare roman numbers of numbers such as 1,4,9,10 etc.

3. Subtract the largest no. that is less than or equal to the given no.

4. Display roman number

Q4. Write a PAC Chart, flowchart, Algorithm to input two complex numbers and add the same to produce the result. After producing the result, print the real part and imaginary part separately.

Data	Process	Output
1. 2 complex numbers	1. input 2 complex numbers 2. Add both of them 3 . show real part 4. show imaginary part	1. sum of 2 complex numbers 2. real part 3. imaginary part

Flowchart:

```
                start

        input complex number

        Sum of complex numbers

            print real part

          print Imaginary part

                 End
```

Algorithm:

1. input 2 complex numbers
2. Add the given numbers
3. Display sum of given numbers
4. Display real part

5. Display imaginary part

Q5. Write a PAC Chart, Algorithm and Flowchart to convert the given integer to the corresponding binary, octal and hexadecimal values and print the same. [CO1] [L1]

Test Cases are

case=1 input=106 output=0b1101010 0o152 0x6a

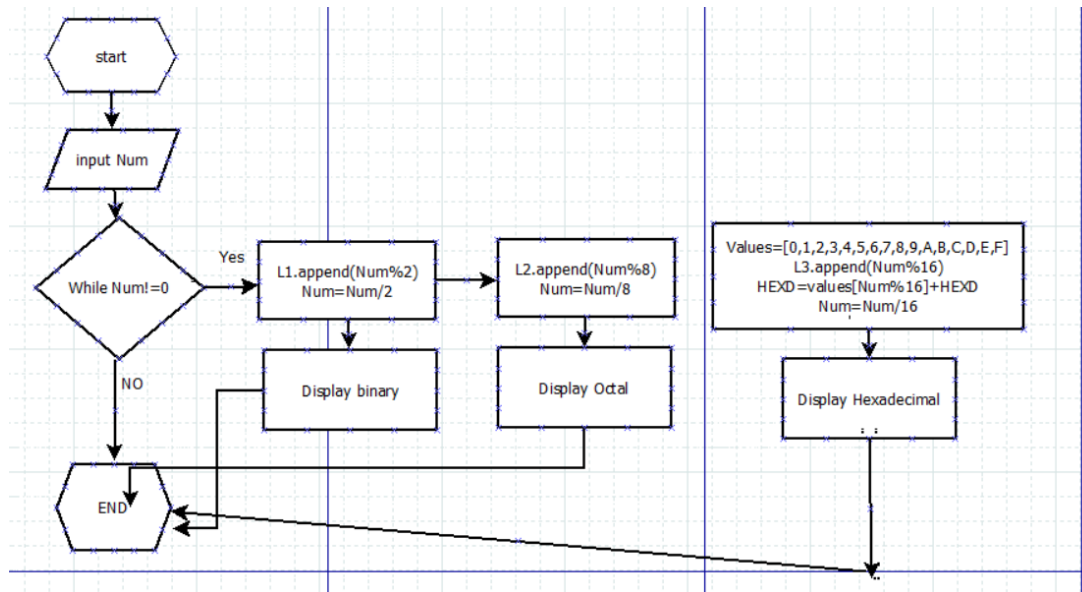case=2 input=98 output=0b1100010 0o142 0x62

case=3 input=9 output=0b1001 0o11 0x9

PAC Chart:

Data	Input	Processing	Output
corresponding binary	106	: Input Number LEN=0 and Y+NUM	0b1101010 0o152 0x6a
octal	98		0b1100010 0o142 0x62
hexadecimal values	9		0b1001 0o11 0x9

Flowchart:

```
        ┌─────────┐
        │  start  │
        └────┬────┘
             ▼
        ┌──────────┐
        │ input Num│
        └────┬─────┘
             ▼
         ◇ While Num!=0 ◇ ──Yes──▶ ┌────────────────────┐   ┌────────────────────┐   ┌─────────────────────────────────┐
                                    │ L1.append(Num%2)   │──▶│ L2.append(Num%8)   │   │ Values=[0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F] │
         NO │                       │ Num=Num/2          │   │ Num=Num/8          │   │ L3.append(Num%16)               │
            │                       └─────────┬──────────┘   └─────────┬──────────┘   │ HEXD=values[Num%16]+HEXD        │
            ▼                                 ▼                        ▼              │ Num=Num/16                      │
        ┌─────────┐                 ┌────────────────┐      ┌────────────────┐       └─────────────────┬───────────────┘
        │  END    │                 │ Display binary │      │ Display Octal  │              ▼
        └─────────┘                 └────────────────┘      └────────────────┘       ┌──────────────────────┐
                                                                                     │ Display Hexadecimal  │
                                                                                     └──────────────────────┘
```

Algorithm:

1. Input NUM
2. While Num !=0
3. append num%2 to L1 and divide num by 2
4. append num%8 to L2 and divide num by 8
5. store hexadecimal values in list called values
6. append num%16 to L3 and add values[num%16] and then divide num by 16
7. display binary,octal,hexadecimal