Assignment -9

Q-1

Ans

```
#include <stdio.h>


int main()
{
  int month;
  printf("  Enter the Month Number i.e 1 to 12  ");
  scanf("%d", &month);


  if (month == 1 || month == 3 || month == 5 || month == 7 || month == 8 || month == 10 || month == 12 )


            printf("\n 31 Days in this Month");


  else if ( month == 4 || month == 6 || month == 9 || month == 11 )


            printf("\n 30 Days in this Month");


  else if ( month == 2 )
  printf("Eithe 28 days or 29 days in this month");
  else
  printf("Enter a valid month number i e from 1 to 12");
  }
```

Q-2

Ans

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

```
{
 int x,a,b;
  while(1)
  {
  printf("\n1. Addition");
  printf("\n2. Subtraction");
  printf("\n3. Multiplication");
  printf("\n4. Division");
  printf("\n5. Exit");
  printf("\nEnter your choice\n");
  scanf("%d",&x);
  switch(x)
  {

    case 1:
    printf("Enter two numbers\n");
    scanf("%d%d",&a,&b);
    printf("Sum is %d ",a+b);
    break;

    case 2:
    printf("Enter two numbers\n");
    scanf("%d%d",&a,&b);
    printf("Difference is %d",a-b);
    break;
    case 3:
    printf("Enter two numbers\n");
    scanf("%d%d",&a,&b);
    printf("product is %d",a*b);
```

```
   break;
   case 4:
   printf("Enter two numbere\n");
   scanf("%d%d",&a,&b);
   printf("Quoten is %d",a/b);
   break;

   case 5:
   break;
   default:
   printf("Invalid choice");

   }
   if(x==5)
   break;

  }//end of loop
    return 0;

}
```

Q-3

Ans

```
#include <stdio.h>

int main()
{
   int day;

   /* Input day number of week from user */
```

```
printf("Enter day number of week\n ");
scanf("%d", &day);

switch(day)
{
   case 1:
      printf("Monday");
      break;
   case 2:
      printf("Tuesday");
      break;
   case 3:
      printf("Wednesday");
      break;
   case 4:
      printf("Thursday");
      break;
   case 5:
      printf("Friday");
      break;
   case 6:
      printf("Saturday");
      break;
   case 7:
      printf("Sunday");
      break;
   default:
      printf("Invalid input! Please enter day number between 1-7.");
}
```

```
    return 0;
}
```

Q-4

Ans

```
#include <stdio.h>
int main()
{
    int sidea, sideb, sidec;
    printf("Input three sides of triangle: ");
    scanf("%d %d %d", &sidea, &sideb, &sidec);

    if(sidea==sideb && sideb==sidec)
    {
        printf("This is an equilateral triangle.\n");
    }
    else if(sidea==sideb || sidea==sidec || sideb==sidec)
    {
        printf("This is an isosceles triangle.\n");
    }
    else
    {
        printf("This is a scalene triangle.\n");
    }

    return 0;
}
```

Q-6

Ans

```
#include<stdio.h>
int main()
{
  int year, remainder;

  printf("Enter Year: ");
  scanf("%d",&year);

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

  switch(remainder)
  {

  case 1:
    printf("Leap Year");
    break;

  case 0:
    printf("Not Leap Year");
    break;

  default:
    printf("Invalid");
    break;

  }

  return 0;
}
```

Q-7

Ans

```
/**
 * C program to calculate total electricity bill
 */

#include <stdio.h>

int main()
{
    int unit;
    float amt, total_amt, sur_charge;

    /* Input unit consumed from user */
    printf("Enter total units consumed: ");
    scanf("%d", &unit);


    /* Calculate electricity bill according to given conditions */
    if(unit <= 50)
    {
        amt = unit * 0.50;
    }
    else if(unit <= 150)
    {
        amt = 25 + ((unit-50) * 0.75);
    }
    else if(unit <= 250)
    {
```

```
        amt = 100 + ((unit-150) * 1.20);
    }
    else
    {
        amt = 220 + ((unit-250) * 1.50);
    }

    /*
     * Calculate total electricity bill
     * after adding surcharge
     */
    sur_charge = amt * 0.20;
    total_amt  = amt + sur_charge;

    printf("Electricity Bill = Rs. %.2f", total_amt);

    return 0;
}
```

Q-10

Ans

```
/**
 * C program to find all roots of a quadratic equation using switch case
 */
include <stdio.h>
#include <math.h> /* Used for sqrt() */

int main()
{
    float a, b, c;
```

```
float root1, root2, imaginary;
float discriminant;

printf("Enter values of a, b, c of quadratic equation (aX^2 + bX + c): ");
scanf("%f%f%f", &a, &b, &c);

/* Calculate discriminant */
discriminant = (b * b) - (4 * a * c);



/* Compute roots of quadratic equation based on the nature of discriminant */
switch(discriminant > 0)
{
   case 1:
      /* If discriminant is positive */
      root1 = (-b + sqrt(discriminant)) / (2 * a);
      root2 = (-b - sqrt(discriminant)) / (2 * a);

      printf("Two distinct and real roots exists: %.2f and %.2f",
            root1, root2);
      break;

   case 0:
      /* If discriminant is not positive */
      switch(discriminant < 0)
      {
         case 1:
            /* If discriminant is negative */
            root1 = root2 = -b / (
```

```
#include <stdio.h>
#include <math.h> /* Used for sqrt() */

int main()
{
   float a, b, c;
   float root1, root2, imaginary;
   float discriminant;

   printf("Enter values of a, b, c of quadratic equation (aX^2 + bX + c): ");
   scanf("%f%f%f", &a, &b, &c);

   /* Calculate discriminant */
   discriminant = (b * b) - (4 * a * c);


   /* Compute roots of quadratic equation based on the nature of discriminant */
   switch(discriminant > 0)
   {
      case 1:
         /* If discriminant is positive */
         root1 = (-b + sqrt(discriminant)) / (2 * a);
         root2 = (-b - sqrt(discriminant)) / (2 * a);

         printf("Two distinct and real roots exists: %.2f and %.2f",
               root1, root2);
         break;

      case 0:
```

```
        /* If discriminant is not positive */
        switch(discriminant < 0)
        {
           case 1:
              /* If discriminant is negative */
              root1 = root2 = -b / (2 * a);
              imaginary = sqrt(-discriminant) / (2 * a);

              printf("Two distinct complex roots exists: %.2f + i%.2f and %.2f - i%.2f",
                    root1, imaginary, root2, imaginary);
              break;

           case 0:
              /* If discriminant is zero */
              root1 = root2 = -b / (2 * a);

              printf("Two equal and real roots exists: %.2f and %.2f", root1, root2);

              break;
        }
    }

    return 0;
}
```