Programs List for if Else

1. Write a C program to find maximum between two numbers.
2. Write a C program to find maximum between three numbers.
3. Write a C program to check whether a number is negative, positive or zero.
4. Write a C program to check whether a number is divisible by 5 and 11 or not.
5. Write a C program to check whether a number is even or odd.
6. Write a C program to check whether a year is leap year or not.
7. Write a C program to check whether a character is alphabet or not.
8. Write a C program to input any alphabet and check whether it is vowel or consonant.
9. Write a C program to input any character and check whether it is alphabet, digit or special character.
10. Write a C program to check whether a character is uppercase or lowercase alphabet.
11. Write a C program to input week number and print week day.
12. Write a C program to input month number and print number of days in that month.
13. Write a C program to count total number of notes in given amount.
14. Write a C program to input angles of a triangle and check whether triangle is valid or not.
15. Write a C program to input all sides of a triangle and check whether triangle is valid or not.
16. Write a C program to check whether the triangle is equilateral, isosceles or scalene triangle.
17. Write a C program to find all roots of a quadratic equation.
18. Write a C program to calculate profit or loss.
19. Write a C program to input marks of five subjects Physics, Chemistry, Biology, Mathematics and Computer. Calculate percentage and grade according to following:
    Percentage >= 90% : Grade A
    Percentage >= 80% : Grade B
    Percentage >= 70% : Grade C
    Percentage >= 60% : Grade D
    Percentage >= 40% : Grade E
    Percentage < 40% : Grade F
20. Write a C program to input basic salary of an employee and calculate its Gross salary according to following:
    Basic Salary <= 10000 : HRA = 20%, DA = 80%
    Basic Salary <= 20000 : HRA = 25%, DA = 90%
    Basic Salary > 20000 : HRA = 30%, DA = 95%
21. Write a C program to input electricity unit charges and calculate total electricity bill according to the given condition:
    For first 50 units Rs. 0.50/unit
    For next 100 units Rs. 0.75/unit
    For next 100 units Rs. 1.20/unit
    For unit above 250 Rs. 1.50/unit
    An additional surcharge of 20% is added to the bill

Write a C program to find maximum between two numbers.

```
/**
 * C program to find maximum between two numbers
 */

#include <stdio.h>

int main()
{
    int num1, num2;

    /*
     * Reads two integer values from user
     */
    printf("Enter two numbers: ");
    scanf("%d%d", &num1, &num2);

    /*
     * Compare num1 with num2
     */
    if(num1 > num2)
    {
        // True part means num1 > num2
        printf("%d is maximum", num1);
    }
    else
    {
        // False part means num1 < num2
        printf("%d is maximum", num2);
    }

    return 0;
}
```

C program to find maximum between three numbers

```
#include <stdio.h>

int main()
{
   int num1, num2, num3, maximum;

   /* Input three numbers from user */
   printf("Enter three numbers: ");
   scanf("%d%d%d", &num1, &num2, &num3);
   if(num1 > num2)
   {
      if(num1 > num3)
      {
         maximum = num1;
      }
      else
      {
         maximum = num3;
      }
   }
   else
   {
      if(num2 > num3)
      {
         maximum = num2;
      }
      else
      {
         maximum = num3;
      }
   }

   /* Prints the maximum value */
   printf("Maximum among all three numbers = %d", maximum);

   return 0;
}
```

Program to find maximum between three numbers using logical operator

```
#include <stdio.h>

int main()

{

   int num1, num2, num3, maximum;

   printf("Enter three numbers: ");

   scanf("%d%d%d", &num1, &num2, &num3);


   /* If num1 is greater than both */

   if((num1 > num2) && (num1 > num3))

   {

      maximum = num1;
```

```
    }
    else if((num2 > num1) && (num2 > num3))
    {
        maximum = num2;
    }
    else if((num3 > num1) && (num3 > num2))
    {
        maximum = num3;
    }

    /* Prints the maximum number */
    printf("Maximum among all three numbers = %d\n", maximum);

    return 0;
}
```

C program to check whether a number is positive, negative or zero

Program to check positive, negative or zero

```
#include <stdio.h>
int main()
{
    int num;
    printf("Enter any number: ");
    scanf("%d", &num);
    if(num > 0)
    {
        printf("Number is POSITIVE");
    }
    else if(num < 0)
    {
        printf("Number is NEGATIVE");
    }
    else
    {
        printf("Number is ZERO");
    }
    return 0;
}
```

C program to check whether a number is divisible by 5 and 11 or not

Program to check divisibility

```
#include <stdio.h>
int main()
{
  int num;
  printf("Enter any number: ");
  scanf("%d", &num);
  if((num % 5 == 0) && (num % 11 == 0))
  {
    printf("Number is divisible by 5 and 11");
  }
  else
  {
    printf("Number is not divisible by 5 and 11");
  }
  return 0;
}
```

C program check whether a number is even or odd

Program to check even or odd

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

```
int main()
{
    int num;
     printf("Enter any number to check even or odd: ");
    scanf("%d", &num);
    if(num%2 == 0)
    {
        printf("Number is Even.\n");
    }
    else
    {
        printf("Number is Odd.\n");
    }
    return 0;
}
```

C program to check Leap Year

Below is the step by step descriptive logic to check leap year:

1. If the year is **exactly divisible** by 4 and **not divisible** by 100, then it is leap year.
2. Else if the year is **exactly divisible** by 400 then it is leap year.
3. If both the condition does not satisfy, then it is a normal year.

Program to check leap year

```
#include <stdio.h>
int main()
{
    int year;
    printf("Enter year : ");
    scanf("%d", &year);
    if(((year % 4 == 0) && (year % 100 !=0)) | | (year % 400==0))
    {
        printf("LEAP YEAR");
    }
    else
    {
        printf("COMMON YEAR");
    }
    return 0;
}
```

C program to check whether a character is alphabet or not

Let us write a step by step descriptive logic to check alphabets

1. Read a character and store in some variable say `ch`.
2. Check if `ch >= 'a' AND ch <= 'z'` or `ch >= 'A' AND ch <= 'Z'`. Then it is an alphabet.

Program to check alphabets

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

```
int main()
{
    char ch;
    printf("Enter any character: ");
    scanf("%c", &ch);
    if((ch >= 'a' && ch <= 'z') || (ch >= 'A' && ch <= 'Z'))
    {
        printf("Character is an ALPHABET.");
    }
    else
    {
        printf("Character is NOT ALPHABET.");
    }
    return 0;
}
```

C program to check vowel or consonant

Program to check vowel or consonant

```
#include <stdio.h>
int main()
{
    char ch;
    printf("Enter any character: ");
    scanf("%c", &ch);
    if(ch=='a' || ch=='e' || ch=='i' || ch=='o' || ch=='u' || ch=='A' || ch=='E' || ch=='I' || ch=='O' || ch=='U')
    {
        printf("%c is VOWEL.", ch);
    }
    else if((ch >= 'a' && ch <= 'z') || (ch >= 'A' && ch <= 'Z'))
    {
        // Condition for consonant
        printf("%c is CONSONANT.", ch);
    }
    return 0;
}
```

Program to check vowel or consonant using ASCII values

```
#include <stdio.h>
int main()
{
    char ch;
    printf("Enter any character: ");
    scanf("%c", &ch);
    if(ch==97 | | ch==101 | | ch==105 | | ch==111 | | ch==117 | | ch==65 | | ch==69 | | ch==73 | | ch==79 | | ch==85
    {
        printf("%c is VOWEL.", ch);
    }
    else if((ch >= 97 && ch <= 122) | | (ch >= 65 && ch <= 90))
    {
        // Condition for consonant
        printf("%c is CONSONANT.", ch);
    }

    return 0;
}
```

C program to check whether a character is alphabet, digit or special character

Program to check alphabet, digit or special character

```
#include <stdio.h>
int main()
{
    char ch;
    printf("Enter any character: ");
    scanf("%c", &ch);
    if((ch >= 'a' && ch <= 'z') | | (ch >= 'A' && ch <= 'Z'))
    {
        printf("%c is ALPHABET.", ch);
    }
    else if(ch >= '0' && ch <= '9')
    {
        printf("%c is DIGIT.", ch);
    }
    else
    {
        printf("%c is SPECIAL CHARACTER.", ch);
    }
    return 0;
}
```

Program to check alphabet, digit or special character using ASCII value

```
#include <stdio.h>
int main()
{
    char ch;
    printf("Enter any character: ");
    scanf("%c", &ch);
    if((ch >= 97 && ch <= 122) || (ch >= 65 && ch <= 90))
    {
        printf("%c is ALPHABET.", ch);
    }
    else if(ch >= 48 && ch <= 57)
    {
        printf("%c is DIGIT.", ch);
    }
    else
    {
        printf("%c is SPECIAL CHARACTER.", ch);
    }
    return 0;
}
```

C program to check whether a character is uppercase or lowercase

Program to check uppercase or lowercase alphabets

```
#include <stdio.h>
int main()
{
   char ch;
   printf("Enter any character: ");
   scanf("%c", &ch);
   if(ch >= 'A' && ch <= 'Z')
   {
      printf("%c is uppercase alphabet.", ch);
   }
   else if(ch >= 'a' && ch <= 'z')
   {
      printf("%c is lowercase alphabet.", ch);
   }
   else
   {
      printf("%c is not an alphabet.", ch);
   }
   return 0;
}
```

Program to check uppercase or lowercase characters using library functions

```
#include <stdio.h>
#include <ctype.h> //Used for isupper() and islower()
 int main()
{
   char ch;
   printf("Enter any character: ");
   scanf("%c", &ch);
   if(isupper(ch))
   {
       printf("%c is uppercase alphabet.", ch);
   }
   else if(islower(ch))
   {
       printf("%c is lowercase alphabet.", ch);
   }
   else
   {
       printf("%c is not an alphabet.", ch);
   }
   return 0;
}
```

C program to enter week number and print day of week

Program

```
#include <stdio.h>
int main()
{
   int week;
   printf("Enter week number (1-7): ");
   scanf("%d", &week);
   if(week == 1)
   {
      printf("MONDAY");
   }
   else if(week == 2)
   {
      printf("TUESDAY");
   }
   else if(week == 3)
   {
      printf("WEDNESDAY");
   }
   else if(week == 4)
   {
      printf("THURSDAY");
   }
   else if(week == 5)
   {
      printf("FRIDAY");
   }
   else if(week == 6)
   {
      printf("SATURDAY");
   }
```

```
    else if(week == 7)
    {
        printf("SUNDAY");
    }
    else
    {
        printf("Invalid Input! Please enter week number between 1-7.");
    }
    return 0;
}
```

C program to enter month number and print number of days in month

Program

```
#include <stdio.h>
int main()
{
   int month;
   printf("Enter month number (1-12): ");
   scanf("%d", &month);
   if(month == 1)
   {
      printf("31 days");
   }
   else if(month == 2)
   {
      printf("28 or 29 days");
   }
   else if(month == 3)
   {
      printf("31 days");
   }
   else if(month == 4)
   {
      printf("30 days");
   }
   else if(month == 5)
   {
      printf("31 days");
   }
   else if(month == 6)
   {
      printf("30 days");
   }
```

```
    else if(month == 7)
    {
        printf("31 days");
    }
    else if(month == 8)
    {
        printf("31 days");
    }
    else if(month == 9)
    {
        printf("30 days");
    }
    else if(month == 10)
    {
        printf("31 days");
    }
    else if(month == 11)
    {
        printf("30 days");
    }
    else if(month == 12)
    {
        printf("31 days");
    }
    else
    {
        printf("Invalid input! Please enter month number between (1-12).");
    }

    return 0;
}
```

Program to print days in a month using logical OR operator

```
#include <stdio.h>


int main()
{
   int month;
   printf("Enter month number (1-12): ");
   scanf("%d", &month);
   if(month==1 || month==3 || month==5 || month==7 || month==8 || month==10 || month==12)
   {
      printf("31 days");
   }
   else if(month==4 || month==6 || month==9 || month==11)
   {
      // Group all 30 days months together
      printf("30 days");
   }
   else if(month==2)
   {
      printf("28 or 29 days");
   }
   else
   {
      printf("Invalid input! Please enter month number between (1-12).\n");
   }
   return 0;
}
```

C program to count total number of notes in given amount

Logic to count minimum number of denomination for given amount

- Read amount in some variable say `amt`.
- Check if the amount is greater than 500.
- Divide the amount by 500 to get maximum number of 500 notes required. Store the quotient in some variable say `note500`.
- Subtract the resultant amount of 500 notes to get final amount. Which is perform `amt = amt - (note500 * 500)`.
- Repeat step 2-4 finding minimum number of notes of 200, 100, 50 and so on.

Program

```
  #include <stdio.h>
int main()
{
    int amount;
    int note500, note100, note50, note20, note10, note5, note2, note1;
    // Initialize all notes to 0
    note500 = note100 = note50 = note20 = note10 = note5 = note2 = note1 = 0;
    /* Read amount from user */
    printf("Enter amount: ");
    scanf("%d", &amount);
    if(amount >= 500)
    {
        note500 = amount/500;
        amount -= note500 * 500;
    }
    if(amount >= 100)
    {
        note100 = amount/100;
        amount -= note100 * 100;
    }
    if(amount >= 50)
    {
        note50 = amount/50;
        amount -= note50 * 50;
    }
    if(amount >= 20)
    {       note20 = amount/20;        amount -= note20 * 20;    }
    if(amount >= 10)
    {       note10 = amount/10;        amount -= note10 * 10;    }
    if(amount >= 5)
    {
        note5 = amount/5;
        amount -= note5 * 5;
    }
    if(amount >= 2)
    {
        note2 = amount /2;
        amount -= note2 * 2;
    }
    if(amount >= 1)
    {
        note1 = amount;
    }
    printf("Total number of notes = \n");
    printf("500 = %d\n", note500);
    printf("100 = %d\n", note100);
    printf("50 = %d\n", note50);
    printf("20 = %d\n", note20);
    printf("10 = %d\n", note10);
    printf("5 = %d\n", note5);
    printf("2 = %d\n", note2);
    printf("1 = %d\n", note1);

    return 0;
}
```

C program to check whether triangle is valid or not if angles are given

Program

```
#include <stdio.h>
int main()
{
    int a, b, c, sum; //a, b, c are three angles of a triangle
    printf("Enter three angles of triangle: \n");
    scanf("%d%d%d", &a, &b, &c);
    sum = a + b + c;
    if(sum == 180 && a!=0 && b!=0 && c!=0)
    {
        printf("Triangle is valid.");
    }
    else
    {
        printf("Triangle is not valid.");
    }
    return 0;
}
```

C program to check whether triangle is valid or not if sides are given

Program

```
#include <stdio.h>
int main()
{
    int a, b, c; //a, b, c are three sides of a triangle
    printf("Enter three sides of triangle: \n");
    scanf("%d%d%d", &a, &b, &c);
    if((a+b) > c )
    {
        if((b+c) > a)
        {
            if((a+c) > b)
            {
                //If a+b>c and a+c>b and b+c>a then it is valid
                printf("Triangle is valid.");
            }
            else
            {
                printf("Triangle is not valid.");
            }
        }
        else
        {
            printf("Triangle is not valid.");
        }
    }
    else
    {
        printf("Triangle is not valid.");
    }
    return 0;     }
```

Program to check valid triangle using nested if

```
#include <stdio.h>
int main()
{
    int a, b, c;
    int valid = 0;
    printf("Enter three sides of triangle: \n");
    scanf("%d%d%d", &a, &b, &c);
    if((a+b)>c )
    {
        if((b+c) > a)
        {
            if((a+c) > b)
            {
                // Change the valid status of triangle to valid
                valid = 1;
            }
        }
    }
    if(valid == 1)
    {
        printf("Triangle is valid.");
    }
    else
    {
        printf("Triangle is not valid.");
    }

    return 0;
}
```

Program

```
#include <stdio.h>
int main()
{
   int a, b, c;
   printf("Enter three sides of triangle: \n");
   scanf("%d%d%d", &a, &b, &c);
   if((a+b > c) && (a+c > b) && (b+c > a))
   {
      printf("Triangle is valid.");
   }
   else
   {
      printf("Triangle is not valid.");
   }
   return 0;
}
```

C program to check whether triangle is valid or not if sides are given

1. Read all three sides of the triangle. Store them in some variable say `a`, `b` and `c`.
2. Check the triangle validity condition. If `a + b > c` and `a + c > b` and `b + c > a`. Then only the triangle is valid, otherwise not.

Program

```
#include <stdio.h>

int main()

{

   int a, b, c;

   printf("Enter three sides of triangle: \n");

   scanf("%d%d%d", &a, &b, &c);

   if((a+b) > c )

   {

      if((b+c) > a)

      {

         if((a+c) > b)

         {

            //If a+b>c and a+c>b and b+c>a then it is valid

            printf("Triangle is valid.");

         }

         else

         {

            printf("Triangle is not valid.");

         }

      }

      else

      {

         printf("Triangle is not valid.");

      }   }

   else

   {      printf("Triangle is not valid.");    }

   return 0;

}
```

Program

```
#include <stdio.h>
int main()
{
    int a, b, c;     int valid = 0;
    printf("Enter three sides of triangle: \n");
    scanf("%d%d%d", &a, &b, &c);
    if((a+b)>c )
    {
        if((b+c) > a)
        {
            if((a+c) > b)
            {
                valid = 1;
            }        }    }
    if(valid == 1)
    {
        printf("Triangle is valid.");
    }
    else
    {
        printf("Triangle is not valid.");
    }    return 0;  }
```

Program

```
#include <stdio.h>
int main()
{
    int a, b, c;
    printf("Enter three sides of triangle: \n");
    scanf("%d%d%d", &a, &b, &c);
    /* If a+b > c and a+c > b and b+c > a then triangle is valid */
    if((a+b > c) && (a+c > b) && (b+c > a))
    {
        printf("Triangle is valid.");
    }
    else
    {
        printf("Triangle is not valid.");
    }
    return 0;
}
```

C program to check whether triangle is equilateral, scalene or isosceles

Program

```
#include <stdio.h>
int main()
{
    int a, b, c
    printf("Enter three sides of triangle: ");
    scanf("%d%d%d", &a, &b, &c);
    if(a==b && b==c)
    {       printf("Equilateral triangle.");    }
    else if(a==b || a==c || b==c)
    {
        printf("Isosceles triangle.");
    }
    else
    {       printf("Scalene triangle.");    }
    return 0;  }
```

C program to find all roots of a quadratic equation

Quadratic equation:

Wikipedia states, in elementary algebra a quadratic equation is an equation in the form of

$$ax^2 + bx + c = 0$$

Solving quadratic equation

A quadratic equation can have either one or two distinct real or complex roots depending upon nature of discriminant of the equation. Where discriminant of the quadratic equation is given by

$$\Delta = b^2 - 4ac$$

Depending upon the nature of the discriminant, formula for finding roots can be given as:

- Case 1: If **discriminant is positive**. Then there are two real distinct roots given by:

$$\frac{-b+\sqrt{\Delta}}{2a} \quad \text{and} \quad \frac{-b-\sqrt{\Delta}}{2a}$$

- Case 2: If **discriminant is zero**. Then it have exactly one real root given by:

$$-\frac{b}{2a}$$

- Case 3: If **discriminant is negative**. Then it will have two distinct complex roots given by:

$$\frac{-b}{2a} + i\frac{\sqrt{-\Delta}}{2a} \quad \text{and} \quad \frac{-b}{2a} - i\frac{\sqrt{-\Delta}}{2a}$$

Based on the above learning let us write a step by step descriptive logic to find roots of a quadratic equation.

1. Read three input from user. Store it in some variable say `a`, `b` and `c`.
2. Find the discriminant of given equation. Use the above formula i.e. `discriminant = (b*b) - (4*a*c)`.
3. Compute the roots based on the nature of discriminant.
4. Check if `discriminant > 0`. Then `root1 = (-b + sqrt(discriminant)) / (2*a)` and `root2 = (-b - sqrt(discriminant)) / (2*a)`
5. Check if `discriminant == 0`. Then `root1 = root2 = -b / (2*a)`.
6. Check if `discriminant < 0`. Then there are two distinct complex roots where `root1 = -b / (2*a)` and `root2 = -b / (2*a)`. Imaginary part is given by `imaginary = sqrt(-discriminant) / (2*a);`.

Program

```
#include <stdio.h>
#include <math.h>
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);
    discriminant = (b*b) - (4*a*c);
    if(discriminant > 0)
    {
        root1 = (-b + sqrt(discriminant)) / (2*a);
        root2 = (-b - sqrt(discriminant)) / (2*a);
        printf("Two distinct and real roots exists: %.2f and %.2f", root1, root2);
    }
    else if(discriminant == 0)
    {
        root1 = root2 = -b / (2*a);
        printf("Two equal and real roots exists: %.2f and %.2f", root1, root2);
    }
    else if(discriminant < 0)
    {
        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);
    }
    return 0;
}
```

C program to calculate profit or loss

Profit = S.P - C.P (Where S.P is Selling Price and C.P is Cost Price)
Loss = C.P - S.P

Program

```
#include <stdio.h>
int main()
{
   int cp,sp, amt;
   printf("Enter cost price: ");
   scanf("%d", &cp);
   printf("Enter selling price: ");
   scanf("%d", &sp);
   if(sp > cp)
   {
      amt = sp - cp;
      printf("Profit = %d", amt);
   }
 else if(cp > sp)
   {
      // Loss
      amt = cp - sp;
      printf("Loss = %d", amt);
   }
   else
   {
      // Neither profit nor loss
      printf("No Profit No Loss.");
   }

   return 0;
}
```

C program to enter student marks and find percentage and grade

Program

```
#include <stdio.h>
int main()
{
    int phy, chem, bio, math, comp; //Five subjects
    float per;
    printf("Enter five subjects marks: ");
    scanf("%d%d%d%d%d", &phy, &chem, &bio, &math, &comp);
    per = (phy + chem + bio + math + comp) / 5.0;
    printf("Percentage = %.2f\n", per);
    if(per >= 90)
    {       printf("Grade A");    }
   else if(per >= 80)
    {       printf("Grade B");    }
    else if(per >= 70)
    {       printf("Grade C");    }
    else if(per >= 60)
    {       printf("Grade D");    }
    else if(per >= 40)
    {       printf("Grade E");    }
    else
    {       printf("Grade F");    }
     return 0;
}
```

C program to enter basic salary and calculate gross salary of an employee

Program

```
#include <stdio.h>
int main()
{
    float basic, gross, da, hra;
    printf("Enter basic salary of an employee: ");
    scanf("%f", &basic);
    if(basic <= 10000)
    {
        da  = basic * 0.8;
        hra = basic * 0.2;
    }
    else if(basic <= 20000)
    {
        da  = basic * 0.9;
        hra = basic * 0.25;
    }
    else
    {
        da  = basic * 0.95;
        hra = basic * 0.3;
    }

    // Calculate gross salary
    gross = basic + hra + da;

    printf("GROSS SALARY OF EMPLOYEE = %.2f", gross);

    return 0;
}
```

C program to calculate electricity bill

Program

```
#include <stdio.h>

int main()

{

    int unit;

    float amt, total_amt, sur_charge;

    printf("Enter total units consumed: ");

    scanf("%d", &unit);

    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);

    }

    sur_charge = amt * 0.20;

    total_amt  = amt + sur_charge;

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

    return 0;

}
```

Switch case programming exercises

1. Write a C program to print day of week name using switch case.
2. Write a C program print total number of days in a month using switch case.
3. Write a C program to check whether an alphabet is vowel or consonant using switch case.
4. Write a C program to find maximum between two numbers using switch case.
5. Write a C program to check whether a number is even or odd using switch case.
6. Write a C program to find roots of a quadratic equation using switch case.
7. Write a C program to create Simple Calculator using switch case.
8.