1. Simple interest

```
#include <stdio.h>
int main()
{
 int P,n;
  float r,si;
   P=1000;
   n=8;
   r = 8.5;
   si = P*n*r/100;
   Print("%f\n", si);
   return 0;
  }
```

3. Conversion of distance

```
 {
  float Km,m,cm,ft,inch;
 Printf ("\nEnter the distance:");
Scanf("%f",&Km);
  m= Km*1000;
  cm = m*100;
  Inch=cm/2.54;
   ft =inch/12:
   Printf("Distance in meters=%f\n", m);
Printf("in centimeter = %f\n", cm);
Printf("in feet = %f\n", ft);
Printf ("in inches=%f\n",inch);
  return 0;
  }
```

2. Gross salary

```
#include <stdio.h>
 int main()
 {
 float bp, da, hra, grpay:
  Printf("\nEnter Basic salary:");
  Scanf("%f",&bp);
   da = 0.4*bp:
   hra = 0.2*bp:
    grpay = bp+da+hra:
  Printf("Basic salary = %f\n";bp)
Printf("Dearness Allowance = %f\n";da)
Printf("House Rent Allowance = %f\n";hra).
Printf("Gross pay of Ramesh %f\n";
 return 0;                 grpay):
 }
```

4. Percentage marks.

```
 {
 int m1,m2,m3,m4,m5,aggr;
 float per;
 Print("\nEnter marks in 5
          sub:");
Scanf("%d %d %d %d %d",&m1,&m2,
   &m3, &m4,&m5);
aggr= m2+m2+m3+m4+m5;
  per = aggr/5;
Print("Agg marks = %d\n", aggr);
Print("per marks = %f\n", per):
return0;

 }
```

5. Temperature into centigrade

```
{
float fahrenheit, celsius;
printf("Enter temperature in
        fahrenheit : ");
scanf("%f", &fahrenheit);
celsius = (fahrenheit - 32) * 5.0/9.0;
Print ("Temperature in
        celsius : %.2f\n", celsius);
return0;
}
```

6. Area, perimeter, circumference

```
{
 float len, breadth, radius;
 float area-rec, peri-rec;
 float area-cir, circum-cir;
 Printf ("Enter length of the rec : ");
 Scanf ("%f", &length);
printf ("Enter breadth of the rec : ");
 Scanf("%f", &breadth);
  Printf ("Enter radius of the cir: ");
 Scanf ("%f", &radius);
   area-rectangle = length * breadth;
Perimeter-rectangle = 2 * (length + breadth);
area - circle = PI * radius * radius;
  circumference- circle = 2 * PI * radius;
 Printf ("Rec Area : %.2f\n", area-rec);
Printf (" Rec peri : %.2f\n", peri-rec);
Printf (" cir Area : %..2f\n", area-cir);
Printf ("cir circum : %..2f\n", circum-cir);
 return 0;
}.
```

7. Dimensions of paper sizes

```
{
 int width = 1189;
  int height = 841;
  Print ("paper sizes A0 to A8:\n");
for (int i= 0; i<= 8; i++){
Print("A%d: %dmm x %d mm\n", i, width,
              heigth):
  if (width > height) {
    width /=2;
   } else {
    heigth /= 2;
  }
 }
 return 0;
}.
```

8. Area of triangle

```
 {
 float a, b, c, sp, area;
 printf ("\nEnter sides of a tri: ");
 Scanf ("%f%f%f", &a, &b, &c);
 Sp = (a+b+c) / 2;
 area = sqrt (sp * (sp - a) * (sp - b) * (sp - c));
 printf ("Area of tringle = %f\n", area);
 return 0;
}
```

9. Total expenses

```
 {int qty, dis;
 float rate, tot;
 printf ("Enter quantity and rate");
 Scanf ("%d %f", &qty, &rate);
 if ( qty > 1000)
      dis = 10;
else
    dis = 0;
  total = (qty * rate) - (qty * rate * dis/100).
Printf ("Total expenses = Rs%f\n", tot);
return 0;
}
```

10. Profit and loss.

```
{
float cp, sp, p, l;
printf ("\nEnter cost price and
   selling price : ");
Scanf ("%f %f", &cp, &sp);
 p = sp - cp;
 l = cp - sp;
 if (p > 0)
  printf (" The seller made a profit
          of RS. %f \n", p);
 if (l > 0)
  printf (" The seller incurred loss of
          RS. %f\n", l);
 if (p = = 0)
  printf (" There is no loss, no profit
                  \n");
 return 0;
}
```

11. Even or odd

```
{int n;
printf ("\nEnter any number: ");
   Scanf ("%d", &n);
   if (n%2 == 0)
  printf (" The number is even\n");
  else
   printf (" The number is odd\n");
   return 0;
 }
```

12. Leap year.

```
{int yr;
printf ("\nEnter a year: ");
Scanf ("%d", &yr);
 if (yr % 100 = 0)
 { if (yr % 400 = = 0)
     printf ("Leap year\n");
else printf ("Not a Leap year\n");
}else
 {
  if (yr % 4 = = 0)
   Printf ("Leap year\n");
else
   Printf ("Not a leap year\n");
}
return 0;
}
```

13. Reverse a number

```
{int num, rN = 0, ON, r;
 Printf (" Enter a 5-digit No: ");
 Scanf ("%d", & num);
  ON = Num;
 while (Num != 0){
     r = num % 10;
    rN = rN * 10 + r;
    num /= 10;
  }
   Printf (" rN : %d\n", rN);
 if (ON = = rN) {
 Printf (" The ON. and rN are equal.
                   \n");
 } else {
 printf (" The ON and rN are not
              equal.\n");
 }
 return 0;
}
```

14. youngest age

```
{int ageRam, ageShyam, ageAjay;
Printf (" Enter the age of Ram: ");
Scanf ("%d", &ageRam);
Printf (" Enter the age of Shyam: ");
Scanf ("%d", &ageShyam);
printf (" Enter the age of Ajay: ");
Scanf ("%d", &ageAjay);
if (ageRam < ageShyam && ageRam <
                  ageAjay){
  Printf (" Ram is the youngest.\n");
}else if (ageShyam < ageRam && age
   Shyam < ageAjay){
```

```
Printf ("Shyam is the youngest.\n");
}else if ( ageAjay <ageRam && ageAjay<
        ageShyam) {
Printf ("Ajay is the yoongest.\n");
}else {
 printf ("There is a tie in age.\n");
 }
 return 0;
 }
```

15. Angles of a triangle

```
{
int angle1, angle2, angle3;
Printf ("Enter the 1st, 2nd and 3rd
        angle of the triangle: ");
Scanf (" %d, %d %d ", &angle1, angle2,
             angle3);
if (angle1 + angle2 + angle 3 == 180) {
  printf (" The triangle is valid.\n");
}else {
Printf(" The triangle is not valid.\n");
}
return 0;
}
```

16. Absolute value

```
{
float number, AV
Printf ("Enter a number: ");
Scanf ("%f , &number);
 if (number <0) {
   AV = -number;
 }else {
   AV = number;
  }
 Printf (" The Absolute value is:
      %.2f\n", AV);
 return 0;
```

17. Area greater than the perimeter

```
{
float length, breadth, area, perimeter;
Printf ("Enter the length of the rectangle: ");
Scanf("%f", &length);
printf ("Enter the breadth of the recta
  -le: ");
Scanf ("%f", & breadth);
  area = length * breadth;
  perimeter = 2 * (length + breadth);
 Printf ("Area of the rec : %.2f\n",
         area);
printf ("perimeter of the rec : %.2f\n",
          perimeter);
if (area >perimeter) {
Printf ("the are of rec is > than its
        perimeter.\n");
}else {
 print("The are of rec is < than its
      perimeter.\n");
 }
 return 0;
 }
```

18. 3 point, straight line

```
{
float x1, y1, x2, y2, x3, y3
float crossproduct;
print ("Enter coordinates of point1
        (x1 y1): ");
 scanf ( "%f %f", &x1 , &y1);
Print ("Enter coordinates of point2
        (x2 , y2): ");
 scanf ("%f %f", &x2 , &y2);
Print ("Enter coordinates of point3
       (x3, y3): "),
 Scanf ("%f %f", &x3 , &y3);
crossproduct= (x2 -x1) * (y3 - y1) -
         (y2 -y1) * (x3-x1);
if (crossproduct ==0){
 print (" the three points \n");
}else {
print (" the three points are not \n")
```

19. Point lies inside, on, on outside.

```
# include <stdio.h>
# include <math.h>
void checkPointPosition(double centerx,
double centery, double radius, double
Pointx, double Pointy) {
double distance = Sqrt(pow(pointx-
centerx, 2) + pow( pointy - centery,2));
if (distance < radius) {
Printf("The point(%.2lf, %.2lf) is
 inside the circle.\n", pointx, pointy);
} else if (fabs (distance radius) < 1e-9) {
Printf("The point (%.2lf, %.2lf) is on
the circle.\n", pointx, pointy);
} else {
printf("The point (%.2lf, %.2lf) is
outside the circle .\n"; pointx, pointy);
}
}
```

20. Point (x, y) lies on the x-axis or y

```
void checkpoint position (double x, double y)
{ if (x == 0 && y == 0) {
 printf ("The point(%.2lf, %.2lf) is at
  the origin.\n", x, y);
} else if ( x == 0) {
 Printf ("The point (%.2lf, %.2lf) is
   on the y axis \n", x, y);
 } else if (y == 0) {
 printf ("The point (%.2lf, %.2lf) is
  on the x-axis.\n", x, y);
 } else {
printf("The point (%.2lf, %.2lf) is
 not on the x-axis, y-axis, or
the origin.\n", x, y);
 }
}
```

21. Day of the week for 1st Jan

```
const char * findDayofweek (int year)
{int day = 1;
int month = 1;
int y = year - (month < 3);
 int m = month + (month < 3) ? 12 : 0);
  int K = y % 100;
 int J = y / 100;
 int f = day + [13 * (m + 1)] / 5 + K + K
  / 4 + J / 4 + 5 * J;
   int dayofweek = f % 7;
const char* daysofweek [] = {
 "Saturday", "Sunday", "Mon",
 "Tue", "wed", "Thur", "Fri"};
return daysofweek [dayofweek];
}
```

22. Salary as per the table

```
 { char g;
 int yos, qual, sal = 0;
 Printf ("Enter Gender, years of
  service and Qualifications (0=G,
  1=PG) : ");
scanf ("%c %d %d", &g, &yos, &qual);
if (g == 'm' && yos >= 10 && qual == 1)
 Sal = 11000;
else if ((g == 'm' && yos >= 10 &&
qual == 0) || (g == 'm' && yos < 10 &&
       qual == 1))
 Sal = 10000;
else if (g == 'm' && yos < 10 && qual == 0)
  Sal = 7000;
else if (g == 'f' && yos < 10 && qual
              == 1)
Sal = 10000;
 Printf ("\nSalary of Employee =
   %d\n", Sal);
```

23. Type of character

```
{char ch;
print("\nEnter a character from
  the keyboard: ");
 scanf ("%c", &ch);
 if (ch>=65 && ch<=90)
printf ("The character in an
       uppercase letter \n");
 if (ch>=97 && ch<=122)
printf ("The character is a lower
       case letter\n");
  if (ch>=48, && ch <=57)
Printf ("The character is digit \n");
 if ((ch>=0 && ch<48) || (ch >57 &&
            ch <65) || (ch>90 && ch<97)
    || ch >122)
printf ("The character is a special
    symbol\n");
return 0;
```

24. Simple interest

```
 { int P, n, count;
   float r, si;
    count =1;
    while (count <=3)
  {
    printf ("\nEnter values of P, n
        and r");
scanf ("%d %d %f", &P, &n, &r);
 si =P * n * r / 100;
  printf ("Simple Interest =Rs.%.\n",
                  si);

    Count = count + 1;

 }
  return 0;

}
```

25. overtime pay.

```
{ float otpay;
 int hour, i=1;
while ( i<=10)

 {printf ("\nEnter no. of hours
               worked: ");
scanf ("%d", &hour);
 if (hour >=40)
    otpay =(hour -40) *120;
else
  otpay =0;
  printf ("Hours = %d overtimepay =
   Rs.%f\n", hour, otpay);
   i++;
}
```

26. Factorial value

```
{ int num, i, fact;
  print("Enter a number: ");
  scanf ("%d, &num);
 fact =i=1;
 while (i<=num)
 {
  fact =fact *i;
   i++;
}
 printf ("Factorial value of
     %d = %d\n", num, fact);
 return 0;

}
```

27. Power

```
{
float x, power;
int y, i;
printf ("\n Enter two numbers
        : ");
Scanf ("%f %d", &x, &y);
  power = i = 1;
  while (i<=y)
  {
   power = power *x ;
   i++ ;
  }
printf (" %f to the power %d
    is %f\n", x, y, power);
  return 0;
}
```

28. Ascii values.

```
{ int i = 0;
printf ("Ascii values : \n");
while (i <= 255) {
  printf (" Ascii value %3d : %c\n"
        , i, i);
  i++ ;
 }
```

29. Armstrong number

```
{int originalNumber, remainder,
    result = 0;
  originalNumber = Number;
while (originalNumber != 0) {
  remainder = ON % 10;
  result += pow(remainder, 3);
  ON /= 10;
  }
  return result == number;
}
```

```
int main() {
printf ("Armstrong numbers
between 1 and 500 are : \n");
for (int i=1; i<=500; i++) {
 if (isArmstrong(i)) {
   printf ("%d\n", i);
   }
 }
 return 0;
}
```

30. Computer's move

```
{int move = 5 - matchsticks % 5;
 return move == 5 ? 1 : move;
}
int main () {
   int matchstick = 21;
   int userpick, computerpick;
printf ("Welcome to the matchstick
      Game!\n");
printf ("There are 21 matchsticks.\n");
printf ("You can pick 1,2,3 or 4
      matchsticks on your turn.\n");
printf ("Whoever picks the last
   matchstick loses the game.\n");
while (matchsticks > 1) {
 printf ("Your turn: pick 1,2,3 or 4: ");
  Scanf ("%d", &userPick);

 }
 if (matchsticks == 1) {
printf ("You are forced to pick
 the last matchstick. You lose!\n");
break;
}
```

31. +ve & -ve numbers.

```
{int number;
 int positivecount =0; negativecount=0;
   zerocount=0;
  char choice;
  do{
  printf("Enter a number:");
   scanf("%d", &number):
 if (number>0){
 positive count++;
 }else if (number<0){
  negativecount++;
  }else{
    zerocount++;
   }
 printf("Do you want to enter
         another number? (y/n):");
 scanf("%c", &choice);
 }while (choice=='y'||choice=='y');
 printf("Count of positive numbers:
      %d\n", positivecount);
printf("count of -ve numbers: %d\n)
printf("count of zeros: %d \n", zero
                count);
  return0;
```

32. Octal equivalent

```
 {int octalNumber(100);
 int i=0;
 if (number==0){
  printf("Octal equivalent: 0\n");
  return;
  }
 while (number!=0){
  OctalNumber[i]=number%0;
  number=number/8; i++;
}
printf("Octal equivalent: ");
for (int j=i-1; j>=0; i++){
 printf("%d, octalNumber[i]);
  }
 printf("\n");
  }
```

33- Diff b/w largest and smallest

```
{int n, number;
int min=INT-MAX;
int max=INT-MIN;
printf("Enter the no-of element:");
scanf("%d; &n);
if (n<=0){
printf("The number of elements should
be greater than 0.\n");
return 1;
 }
 printf("Enter the number:\n");
for (int i=0;i<n; i++){
Scanf("%d", &number);
 if (number<min){
   min=number;
  }if (number>max){
       max=number;
    }
   }int range = max-min;
Printf(".The smallest number is:
        %d\n", min);
Printf("The biggest is: %d\n", max)
 printf("The range is: %d\n", range
return 0;
}.
```

34 · prime or not

```
{ int num, i;
 Printf ("Enter a number");
  Scanf ("%d", & num);

  i=2;
 while (i<=num-1)
 { if (num %i==0).
  {
    Printf ("Not a prime number.");
   break
  }
   i++;
  }
  if (i==num)
      printf ("Prime nomber\n");
 }
```

35. Prime Nombers.

```
{int i, n=1;
Printf ("\n Prime numbers between
        1and 300are:\n 2\t");
 for (n=1; n<=300; n++)
  {
   i=2;
  for (i=2; i<n; i++)
   {
    if (n%i==0)
     break;
   }
    if (i==n)
  printf ("%d\t", n);
  }
   retorno;
  }
```

36. Sum of 1st seven terms of a Series.

```
{ int i=1, j;
float fact, sum=0.0;
 for (i=1; i<=7; i++)
 {
 fact=1.0;
 for (j=1; j<=i; j++)
 fact=fact*j;
  sum=sum+i/fact;
 }printf ("sum of series = %f\n", sum);
 retorno;
   }
```

37. Combinations.

```
{int i=1, j=1, k=1;
 for (i=1; i<=3; i++)
 { for (j=1; j<=3; j++)
  { for (k=1; k<=3; k++)
      printf ("%d%d%d\n", i, j, k);
 }
 }
```

38. Multiplication table

```
{
 int nomber;
 printf ("Enter the nomber: "),
 scanf("%d", &nomber);
 printf ("Multiplication table of %d:
    \n", nomber),
  for (int i=1; i<=10; i++){
 printf("%d *%d =%d\n",
    nomber, i, nomber *i);
   }
```

39. factorial value

```
{
 int num;
 int factorial;
 Printf ("\nEnter a number: ");
 Scanf ("%d", &num);
 factorial = fact(num);
 printf ("Factorial of %d = %d\n",
   num, factorial);
   return 0;
 }
int fact (int num)
 {
  int i;
  int factorial=1;
  for (i=1; i<=num; i++)
  factorial = factorial * i;
  return(factorial);
  }
```

40. power of a value.

```
 {
 float x, pow;
 int y :
 printf ("\nEnter two numbers");
  Scanf ("%f %d", &x, &y);
  pow = power (x, y);
 printf ("%f to the power %d = %f\n",
 x, y, pow);
 return 0;
 }
  float power (float x, int y)
  {
   int i;
   float p=1;
   for (i=1; i<=y; i++)
    p = p * x;
   return (p);
     }
```

41. year into its roman equivalent

```
 { int yr;
  printf ("\nEnter year: ");
 Scanf ("%d", &yr);
  yr = romanise (yr, 1000, 'm');
  yr = romanise (yr, 500, 'd');
  yr = romanise (yr, 100 'c');
  yr = romanise (yr, 50, 'l');
  yr = romanise (yr, 10, 'x');
  yr = romanise (yr, 5, 'v');
  romanise (yr, 1, 'i');
   return 0;
  } int romanise (int y, int k, char ch)
 { int i, j;
  j = y / k;
  for (i=1; i<=j; i++)
      printf ("%c", ch);
     return (y % k);
    }
```

42. Sum, average and standard deviation.

```
{ int sum, avg;
 double stdev;
  stats (&sum, &avg, &stdev);
 Printf ("Sum = %d\nAverage = %d
 \nStandard deviation = %lf\n",
     sum, avg, stdev);
      return 0;
  }
 void stats (int *sum, int *avg,
double *stdev)
{
 int n1, n2, n3, n4, n5;
 Printf("\nEnter 5 numbers: ");
Scanf("%d%d%d%d%d", &n1, &n2, &n3,
     &n4, &n5);
 *sum=n1+n2+n3+n4+n5;
 * avg= *sum/5;
*stdev = sqrt((pow((n1-*avg),2.0)
  +pow((n2-*avg),2.0)+(pow
 ((n3-*avg), 2.0)+pow((n4-*avg),
 2.0)+ (pow((n5-*avg),2.0))/4);
 }
```

43. Power and factorial value.

```
{
 float a;
 int b, number, factorial;
  float pow;
printf("Enter a and b for calculating
    a raised to b: ");
Scanf("%f %d", &a, &b);
printf("Enter number whose
 factorial is to be calculated:");
Scanf("%d", &number);
power_fact(a,b,number,&pow,&
     factorial);
printf("power = %f Factorial = %d"
  pow, factorial);
 return0;
}
void power_fact(float x, int y,
  int num, float *power, int *fact)
{
 float res=1;
   int i;
for(i=1;i<=y;i++)
  res=res*x;
 *power=res;
  res=1;
 for(i=1;i<=num;i++)
  res=res*i;
   *fact=res;
  }
```

44. Memory map

```
{
 float x = 3.14;
 float *y;
 float **z;
  = &x;
  z = &y;
 printf("%p %p %p\n", &x, &y,
        &z);
Printf("%p %p\n", y, z);
Printf("%f %f %f", x, *y, **z);
return 0;
}.
```

45. Circular right shift

```
void Circularrightshift(int *a,
        int *b, int *c){
 int temp = *c;
  *c = *b;
 *b = *a;
  *a = temp;
 }
 int main(){
  int x = 5, y=8, z=10;
 print("Before circular right
         shift :\n");
printf("x = %d, y = %d, z = %d\n",
       x, y, z);
Circulanrightshift (&x, &y, &z);
Putntf ("After circular aright
      shift : \n");
Pntntf ("x = %d , y = %d, z = %d \n",
      x, y, z);
 Retunn 0;
}
```

46. Convensition of weight

```
void convertweight(double kilograms,
double * grams, double * tons, double
 * pounds){
* grams = kilograms * 1000;
* tons = kilogram / 1000 ;
 * pounds = kilograms * 2.20462 ;
}
int main(){
 double Kilograms, grams, tons,
   pounds;
 Putntf ("Enten the weight in kg");
 Scanf ("%lf", &Kilograms);
 Convertweight (kilograms, &grams,
  &tons, &pounds);
 pnint f ("Equivalent weight in:\n");
 pntntf ("Grams : %.2lf\n", grams);
pnintf ("Tons : %.2lf\n", tons);
pnintf ("Pounds :%.2lf\n", pounds);

 Retunn 0;

}.
```

47. Sum of digits

```
{int num, sum;
  int n;
 pnintf ("Enten number: ");
Scanf ("%d", &num);
 Sum = nsum (num);
  pnintf("sum of digit is %d\n", sum);
    Retunn0;
 }
 int nsum (int n)

 {
  int s, nemainden;
   if (n!=0)
  { nemainden = n % 10;
     S = nemainden + nsum(n /10);
  }
   else
   netunn 0;
   netunns;
 }.
```

48. prime factors

```
{ int num;
 pnintf ("Enten a number: ");
 scanf ("%d", &num);
  pnintf ("prime factons are : ");
   factonize(num, 2);
    netunn 0;
 }
 void factonize (int n, int i)
  { if (i<=n)
   { if (n % i == 0)
    {
     pnintf ("%d, i);
     n = n/i;
    }
else
  i++;
  factorize(n, i);
}
}
```

49. first 25 terms of fibonacci sequence.

```
{int terms = 25, i, n=0;
 for (i=1; i<=terms; i++)
 {
  printf("%d\t", fibo(n));
   n++;
 }
 return0;
}
int fibo(int n)
{ if (n==0||n==1)
   return n;
   else
  return (fibo(n-1)+fibo(n-2));
}.
```

50. operations.

```
{
 int arr[] = {10, 20, 30, 45, 67,
                      56, 74};
 int i = 4, *j, *k, *x, *y;
   j = &i;
   j = j+9;
  k = j+9;
   k = &i;
   k = k-3;
   x = &arr[1];
   y = &arr[5];
    print ("%d\n", y-x);
    j = &arr[4];
  k = (arr+4);
  if (j==k)
  printf("The two pointers
  point to the same location\n");
else
 printf("The two pointers point
 to different locations\n");
   return0;
  }.
```