16-05-24

Q:- Finding sum of individual digits of given number:-

```
-> main()
  {
   int m, n, s=0;
   printf("enter n");
   scanf("%d",&n);
     while(n!=0)
      {
        m=n%10;
        s=s+m;
        n=n/10;
      }
     printf("%d", s);
  }
```

```
m=n%10;
s=s+m;
n=n/10;
```

Q:- WAP to reverse a number;-

```
-> main()
  {
   int m, n, s=0;
   printf("enter n");
   scanf("%d", &n);
    while(n!=0)
    {
      m=n%10;
       s=s*10+m;
        n=n/10;
    }
   printf("%d", s);
  }
```

```
m=n%10;
s=s*10+m;
n=n/10;
```

Q:- WAP to check given number is Palindrome or not :-

→ main ( )

```
{ int m, n, s=0, x;
  printf ("enter n");
  scanf ("%d", &n);
    x = n;
    while ( n != 0)
    {
      m = n % 10;
      s = s * 10 + m;
      n = n / 10;
    }
    if ( x == s)
    printf ("Palindrome");
    else
    printf ("Not a Palindrome");
}
```

```
x = n;
while(n != 0)
{m = n % 10;
 s = s * 10 + m;
n = n / 10;
 }
 if (x == s)
```

Q:- WAP to check given number is Armstrong or not :-

→ main ( )

```
{ int m, n, s=0, x;
  printf ("enter n");
  scanf ("%d", &n);
    x = n;
    while (n != 0)
    {
      m = n % 10;
      s = s + m * m * m;
      n = n / 10;
    }
    if (x == s)
    printf ("Armstrong");
    else
    printf ("Not an Armstrong");
}
```

```
x = n;
while (n != 0)
{
m = n % 10;
s = s + m * m * m;
n = n / 10;
}
if (x == s)
```

Q:- WAP to check the given number is prime or not :-

```
main()
{
int i, n, c=0;
printf("enter n");
scanf("%d", &n);
i=2;
while(i<=n/2)
{
  if(n%i==0)
  c++;
  i++;
}
if(c==2)
printf("Prime");
else
printf("Not Prime.");
}
```

```
i=2
while(i<=n/2)
{
  if(n%i==0)
  c++;
  i++;
}
if(c==2)
  prime;
else
  not prime;
```

```
i=2;
while(i<=n/2)
{ if(n%i==0)
  c++;
  i++;
}
if
```

Q:- WAP to check the given number is perfect or not :-

```
main()
{
int i, n, s=0;
printf("enter n");
scanf("%d", &n);
i=1;
while(i<=n/2)
{
  if(n%i==0)
  s=s+i;
  i++;
}
if(s==n)
printf("Perfect");
else
printf("Not perfect");
}
```

```
i=1;
while(i<=n/2)
{ if(n%i==0)
  s=s+i;
  i++;
} if(s==n)
perfect; else not perfect;
```

Q:- Neon Number

$9^2 = 81 = 8 + 1 = 9$

```
-> main ()
{ int n, sq, rem, sum = 0;
  printf ("enter n");
  scanf ("%d", &n);
  sq = n * n;
  while (sq != 0)
  { rem = sq % 10;
    sum += rem;
    sq /= 10;
  }
  if (sum == n)
  printf ("Neon No");
  else
  printf ("Not a Neon No");
}
```

9 * 9
sum +

81 / 10 = 8 (1)
8 / 10 = 0 (8)

sum += rem;
sum = sum + rem;

* all programs write in do-while

Q:-

```
-> main ()
{ int a;
  printf ("enter a");
  scanf ("%d", &a);
  do
  { printf ("%d", a);
    a++;
  }
  while (a <= 10);
}
```

if-else
1 -> Palin
2 -> Armn

if - else
1 - Palindrome
2 - Armstrong

```
main()
{
int m, n, s=0, x, a, y;
printf("enter n");
scanf("%d", &n);
printf("enter 1 for Palindrome 2 for Armstrong");
scanf("%d", &y);
if(y==1)
{
x = n;
while(n!=0)
{
m = n%10;
s = s*10+m;
n = n/10;
}
if(s==x)
printf("Palindrome");
else
printf("Not a Palindrome");
}
else if(y==2)
{
x = n;
while(n!=0)
{
m = n%10;
s = s+m*m*m;
n = n/10;
}
if(x==s)
printf("Armstrong");
else
printf("not Armstrong");
}
}
```

y =>
1 - P
2 - Not A

☆ All programs Write in do-while()

do-while ( ) :-

Q:- Finding sum of individual digits of given number :-

-> int main ( )

```
{ int m, n, s = 0;
  printf ("enter n:");
  scanf ("%d", &n);
  do {
      m = n % 10;
      s = s + m;
      n = n / 10;
     }
     while (n != 0)
     printf ("sum of individual digits is: %d\n", s);
}
```

Q:- WAP to reverse a number :-

-> int main ( )

```
{
  int m, n, s = 0;
  printf ("Enter n: ");
  scanf ("%d", &n);
  do {
      m = n % 10;
      s = s * 10 + m;
      n = n / 10;
     }
     while (n != 0);
     printf ("the reversed number is: %d \n", s);
}
```

Q:- WAP to check given number is Palindrome or Not :-

→
```
main()
{
    int n, x, m, s=0;
    printf("Enter n: ");
    scanf("%d", &n);
    x = n;
    do {
        m = n % 10;
        s = s*10 + m;
        n = n / 10;
    }
    while (n > 0);
    if (x == s)
    printf("Palindrome");
    else
    printf("Not a Palindrome");
}
```

Q:- WAP to check given number is armstrong number or not :-

→
```
main()
{
    int n, x, m, s=0;
    printf("Enter n: ");
    scanf("%d", &n);
    x = n;
    do {
        m = n % 10;
        s = s + m*m*m;
        n = n / 10;
    }
```

```
while (n > 0);
if (x == s)
printf ("%d is an Armstrong number", x);
else
printf ("%d is not an Armstrong number", x);
}
```

Q:- WAP to check the given number is prime or not :-

→

```
main ()
{ int i, n, c=0;
  printf ("enter n:");
  scanf ("%d", &n);
  i=2;
  do{
     if ((n!=2) && (n%i==0))
     {
       c=1;
       break;
     }
     i++;
  }
  while (i<=n/2);
  if (c==0)
  printf ("%d is a Prime number", n);
  else
  printf ("%d is not a Prime number", n);
}
```

Q:- WAP to check the given number is perfect number or not :-

```
-> main()
{
  int i, n, s=0;
  printf("Enter n: ");
  scanf("%d", &n);
  i=1;
  do {
      if (n%i==0)
      s=s+i;
      i++;
    }
  while (i<=n/2);
  if (s==n)
  printf("%d is a Perfect number", n);
  else
  printf("%d is not a Perfect number", n);
}
```

Q:- WAP to check the given number is neon number or not :-

```
-> main()
{
  int n, sq, sum=0;
  printf("Enter n: ");
  scanf("%d", &n);
  sq=n*n;
  do {
      sum+=sq%10;
      sq/=10;
    }
  while(sq!=0);
  if (sum==n)
  printf("%d is a neon number", n);
  else
  printf("%d is not a neon number", n);
}
```

0 & 9 are Neon No's

Q:- Program to check if a number is both a Palindrome and an Armstrong number:-

→
```
#include <stdio.h>
#include <math.h>
int main()
{
    int n, original_num, reversed_num = 0, remainder, sum = 0,
    num_digits = 0;

    // input number from user
    printf("Enter an integer: ");
    scanf("%d", &n);

    original_num = n;

    do {
        num_digits++;
        n /= 10;
    } while (n != 0);

    n = original_num;

    do {
        remainder = n % 10;
        reversed_num = reversed_num * 10 + remainder;
        sum += pow(remainder, num_digits) // Add the
    digits to raised to num_digits to the sum
        n /= 10; // Remove the last digit from the number
    }
    while(n != 0);    // Continue until 'n' becomes 0
```

```
// Check if the number is Palindrome
if (original_num == reversed_num)
{
  printf ("%d is a palindrome number \n", original_num);
}
else
{
  printf ("%d is not a palindrome number \n", original_num);
}
// check if the number is Armstrong
if ( sum == original_num)
{
   printf ("%d is an armstrong number \n", original_num);
}
 else
 {
   printf ("%d is not an armstrong number \n", original_num);
 }
}
```