}

20/03/17

FUNCTIONS

function declaration :

return-type f_name(list of datatype ~~of~~ parameters) ;

```
int f (int, char);
```

function definition :

return-type f_name (list of data type ~~of~~ with parameters name)

```
int f (int x, char ch)
```

function call :

$\boxed{\text{var} =}$ f (var/const/exp);

(other than void)

22/03/17

Q. WAP to input 2 no. calculate their sum using func and print the result in the main function.

```
int sum (int , int )
void main( )
{
    int a , b , sum1
    ~~sum1 = sum (a~~
    scanf("%d %d", &a, &b);
    sum1 = sum (a , b);          | printf("%d", sum(a,b));
    printf("%d", sum1);          |
}

int sum (int x , int y )
{
    int temp ;                 |
    temp = x + y;              |   return (x+y);
    return (temp);             |
}
```

Q. WAP to calculate the following expression $\frac{\lfloor a}{\lfloor b \lfloor c}$

```
~~float~~ int fac (int); ~~int , int );~~
void main( )
{ float res
  int a , b , c , ~~res~~ fa , fb , fc
  scanf("%d %d %d", &a, &b, &c);
  ~~printf("%d", fac(a, b, c));~~
}   fa = fac (a);
    fb = fac (b);
    fc = fac (c);
    res = fa /(fb * fc);
    printf("%d", res);
```

```
(x)'
        int
  ~~float~~ fac(int x) ~~int y, int z~~)
 x^3 <- {
        int f=1, i;
        for (i= 2; i<=x; i++)
        {
           f=f*i;
         }
       return(f);
     }
```

Q. WAP to input a +ve integer no. and by using a function check whether it is a pallandrome or not. If it is a pallandrome then print message in main function

```
  int pal(int);
 void main()
 {
   int num, res
   scanf("%d",&num);
 res = pal(num);
   if(res==1)
   { printf("pal");
   }
   else
   { printf("NAP");
   }
  getch();
 }
```

```
int pal (int x )
{
    int sum = 0 , temp
      temp = x;
   while (x > 0)
   {
      sum = sum * 10 + x % 10;
          x = x /10 ;
   }
 if (sum == temp)
  {   return (1);
  }
  else
   {  return (0);
  }
}
```

27/03/17

Q. WAP to implement power function for +ve integer exponent and base without using pow() function.

pow()

```
int powr(int, int)
void main( )
{
   int res, a, b;
   scanf("%d %d", &a, &b);
   res = powr(a, b);
   printf("%d", res);
  getch();
}
 int powr(int x, int y)
{ int i, p = 1;
  for(i=Φ, i<=y; i++)
  {
       p = p* x;
  }
  return(p);
}
```

Q. WAP to calculate the sum of following series:

$x + x^3 + x^5 + x^7 + \ldots x^n$

Use function input to input the values, use function output to print the result, use function power to calculate powers.

```
int input ( ~~[illegible]~~ )
int power ( int , int )
void output (int)
void main( )
{
  ~~[illegible]~~ ~~[illegible]~~ int x, n, i;
  ~~[illegible]~~ float sum=0;

 x = input(~~n~~);
 n = input( );

~~int input([illegible])~~
~~{ int n;~~
  ~~scanf("%d", n)~~

for(i=1 ; i<=n ; i+=2)
 {
    sum += power(x, i);
 }

 output(sum);
 getch( );
}
int input(~~int n~~)
{ int num
  scanf("%d", &num);
  return(num);
}

void output(int res)
 { printf("%d", res);
 }
```

```
int power(int b, int c)
{ int i, p=1;
for(i=1 ;i<=c ; i++)
 { p=p*b;
}
return(p );
}
```

28/03/17

Q. WAP to input 2 no. and by using function swap these two no.

```
 int a , b ;
void swap ( ~~int~~ ~~int~~ );
 void main ()
 {
   ~~int a , b ;~~
   scanf ("%d %d", &a, &b);
   swap ( ~~a, b~~ );
   ~~getch ( );~~
   printf ("%d %d", a, b);
   getch ( );
 }
void swap ( ~~int a~~ , ~~int b~~)
 {
     int temp;
     temp = a ;
     a = b ;
     b = temp;
 }
```

TYPES of FUNCTION on the basis of arguement and return type.

1. No returntype No arguements
2. No returntype with arguements
3. With return type but no arguements.
4. With return type with arguements
5. function which can return multiple values.

Q. Add two numbers.

①
```
void sum();
void main()
{
    sum();
    getch();
}
void sum( )
{
    int a, b, res;
    scanf("%d %d", &a,&b);
    res = a + b;
    printf("%d", res);
}
```

②
```
void sum(int, int);
void main()
{
    int a, b;
    printf("enter values");
    scanf("%d %d", &a,&b);
    sum(a, b);
    getch();
}
void sum(int x, int y)
{   printf("%d", x+y);
}
```

③
```
int sum ( );
void main ()
{
   int res ;
   res = sum( );
   printf("%d", res);
  getch();
}
int sum( )
{
   int a , b;
  printf ("enter values);
  scanf ("%d%d",&a,&b);
  return (a+b);
}
```

④
```
int sum (int, int );
void main( )
{
   int a , b ;
   scanf ("%d%d", &a , & b);
   printf(" %d ", sum(a ,b));
  getch();
 }
int sum (intx , int y)
{
   return(x+y);
}
```

Q. WAP to calculate the following $\exp^n$

$$^nC_r = \frac{n!}{r!(n-r)!}$$

1
```
void calculate();
int fact (int );
int input ();
void output(float);
void main( )
{

     calc( );
    getch ( );
  }
void calc( ) ;
 {    int n , r;
res = [illegible] (fac(n)/(fac(r) * fac(n-r)));
      output (res);
 }

int fac (int x )
  {      int n , r , f=1 , i ;
    n= input ( );
    r= input ( ) ;
        for (i = 2 ; i <= x ; i++ )
        {
             f = f * i
        }
        return (f) ;
  }
int input ( )
 {      int x
    scanf ("%d", &x );

 }
void output ( float r)
 {  printf ("%f", r);

 }
```

```
(2)  void calculate()
     .int input ()
      void output(float);
      int fact(int);
     void main()
     {
        calculate();
        getch();
     }

     void calculate()
     {
        int n, r
        float res;
        n= input();
        r= input();
        res= fac(n)/(fact(r) * fact(n-r));
        output(res);
      }
      int input()
      {
        int x;
        scanf("%d", &x);
        return (x);
      }
      void output (float r)
      {  printf("%f", r);
      }
      int fact (int num )
      {
        int f=1, i;
        for(i= 2; i<=num; i++)
        {    f=f* i;
        }
        return (f);
      }
```

How we pass array to a function

How we pass pointers to a function:

Q. WAP to input two no. then pass the address of these two no. to the function and print the result in the main function.

```
int sum(int * , int * )
void main()
{
  int a, b;
  scanf("%d %d", &a, &b);
  printf("%d", sum(&a, &b));
  getch();
}
int sum(int *p , int *q)
{
  return(*p + *q);
 }
```

Q. WAP to input two no. and then by using a single function calculate addition & subtraction of these two no. and print the result in the main function

```
  void calculate(int, int )
① int sum, sub;
  void main()
  { int a, b;
    scanf("%d %d", &a, &b);

    calculate (a , b);
    printf("%d%d", sum, sub);
    getch();
  }  void calculate (int x , int y)
  {   sum = x+y ;
      sub = x-y ;
```

②
```
void calc(int, int, int*, int*);
void main()
{  int a, b, sum, sub;
   scanf("%d %d", &a, &b);
   calc(a, b, &sum, &sub);
   printf("%d %d", sum, sub);
   getch();
}
void calc(int x, int y, int *p, int *q)
{
      *p = x + y;
      *q = x - y;
}
```

Passing array through a function.

(int [ ], int) | (int [ ], int, int)

↳ size

Returntype f-name (datatype with [ ]);

f_name (name of array);

```
Returntype f-name (datatype with parameter [ ])
{

   }
```

```
void f(float [ ]);
void main()
{   float a[5];
        f(a);        ——→  call by reference
}
```

```
int f(float b[])
{
   }
```

Returntype f-name(datatype *);

f-name(name of array/& array[0]);

```
Returntype f_name(datatype *parameter);
{
   }
```

```
int f(float * );
void main()
{
     float a[5];
         f(a);
}
int f(float *p).
{
    }
```

Q. WAP to input n no. in array then pass elements of this array to a function and print factorial of all numbers.

(1)
```
void fact (int[], int);
void main ()
{
      int a[50], n, i;
      scanf("%d", &n);
      for (i=0; i<n; i++)
      {  scanf("%d", &a[i]);

      }
```

```
        getch();
      }

    void fact(int a[], int n)
    {
        int i, j, f=1;
        for(i=0;i<n;i++)
        {     f=1;
              for(j=2;j<=a[i];j++)
              {     f=f*j;
              }
              printf("%d",f);
        }
      }
```

②

```
int fact(int);
void main()
{
    int i, n, a[20];
    scanf("%d", &n);
    for(i= 0;i<n;i++)
    {
      scanf("%d", &a[i]);
    }
    for(i= 0;i<n;i++)
    {   fact(a[i]);
    }
    getch();
}
```

```
void fact (int x)
{  int i, f=1;
  for(i=2; i<=x ; i++)
  {
     f=f* i;
  }
  printf("%d", f);
}
```

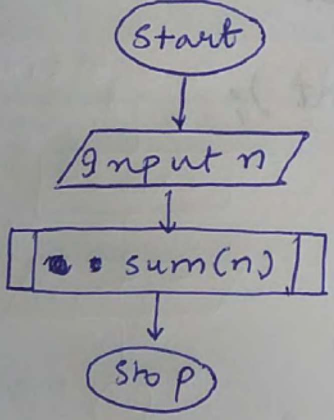
Q. WAP to input ~~n~~ no. in an array and then by using a function sort that array.

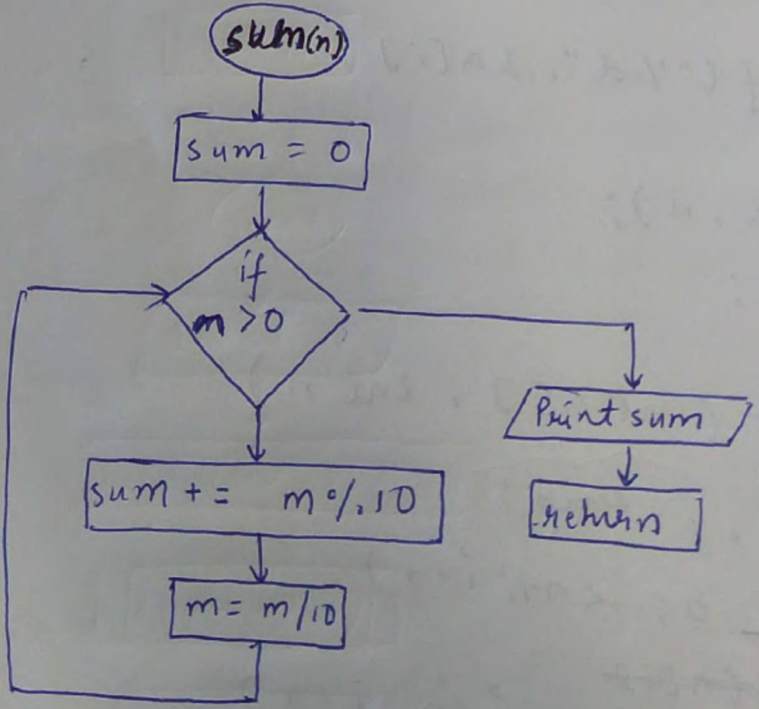
```
 void sort (int [ ], int );
 void main( )
 {
    int a[50], i , n ,
    scanf("%d", &n);
   for(i=0; i<n; i++)
   {
     scanf("%d", &a[i] );
   }
   sort (a, n);
   getch( );
 }
 void sort (int a[ ] , int n)
 {   int i, j, temp;
    for (i= 0; i<n-1; i++)
    {    ~~if(a[i]~~
       for(j=i+1; j<n; j++)
        { if (a[i] > a[j]
          {  temp = a[i];
```

~~a[0]=a[~~

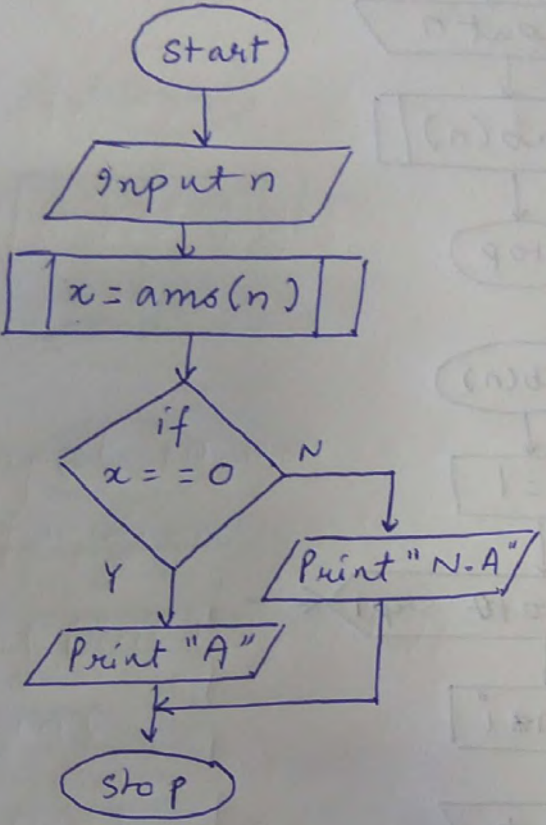
```
            a[i] = a[j];
            a[j] = temp;
        }
    for(i=0; i<n; i++)
    {
        printf("%d", &a[i]);
    }
}
```

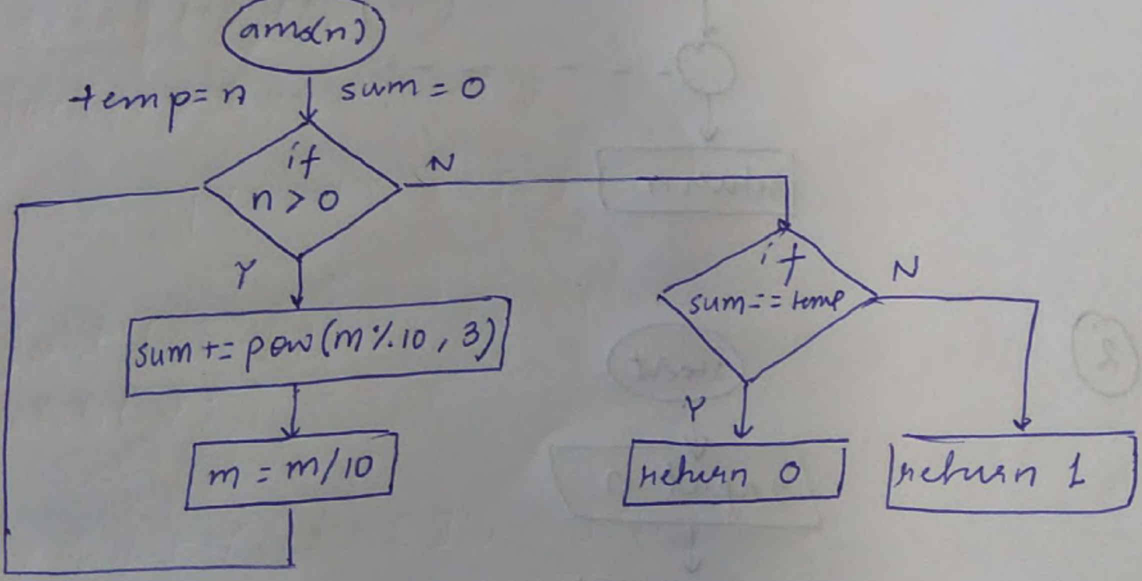
DAF and by using a f^n calculate sum of digit of a no.





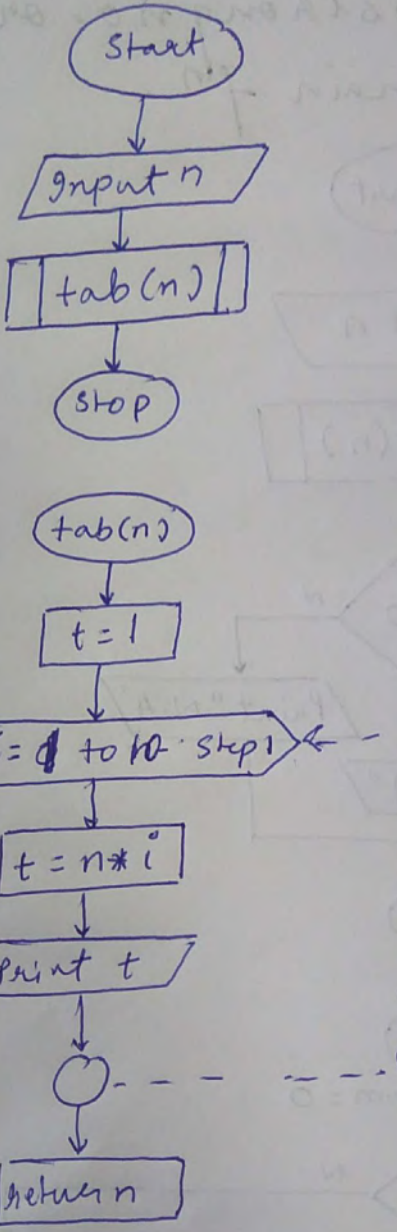
1) Af to input a no. and by using a f^n check whether its a amstrong no. or not. Print the message in the main f^n.



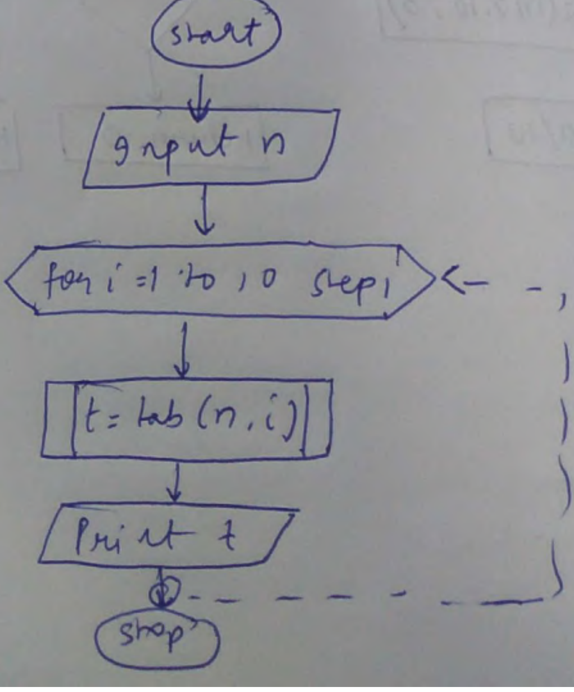


DAf to input a no. and print its table using fⁿ

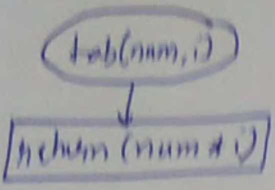
1



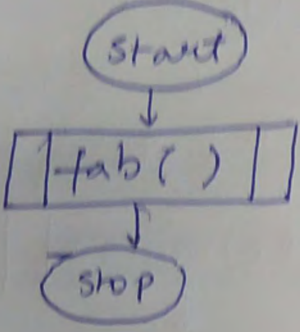
2

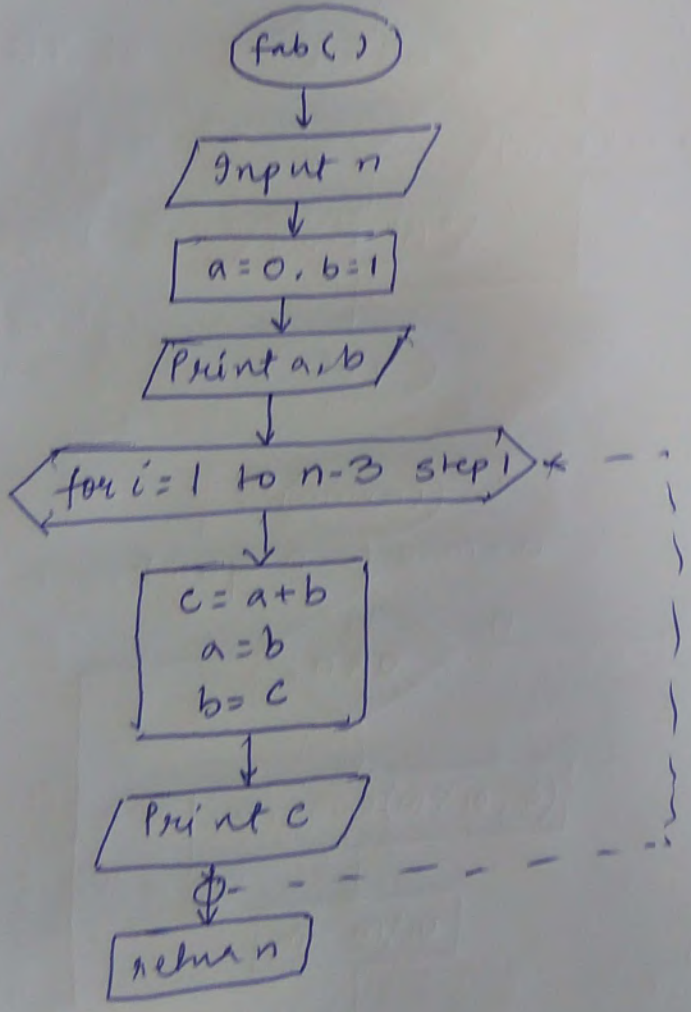


Scanned by CamScanner



DAf to print fabunacci series using function.





10/04/17

Recurssion

* calling itself in its definition
* should have a termination condition

Q. WAP to calculate factorial of a no. using recurssion f^n.

```
int fact (int);
void main()
{
    int n;
    scanf ("%d", &n);
    printf ("%d", fact(n));
    getch ();
}

int fact (int x)
{
    if (x==0 || x==1)
        return (1);
    else
        return (x *fact(x-1));
}
```

Q. WAP to calculate sum of following series
1 + 2 + 3 + . . . + n

```
int sum (int);
void main ()
{
    int n;
    scanf ("%d, &n);
        print ("%d", sum(n));
        getch ();
```

Scanned by CamScanner

```
int sum(int x)
{
  if(x== 1)
      return 1;
  else
   return (x + sum(x-1));
}
```

Q. $1^2 + 2^2 + 3^2 + 4^2 + \dots + n^2$

```
int sum(int);
void main()
{
  ~~int main()~~
  int n;
  scanf("%d",&n);
  printf("%d", sum(n));
  getch();
}
int sum(int x)
{
  if(x==1)
      return 1;
  else
    return(pow(x,2) + sum(x-1));
}
```

$4^2 + \underbrace{\cancel{sum}(3)}_{14}$

$3^2 + \underbrace{\cancel{sum}(2)}_{5}$

$2^2 + \underbrace{\cancel{sum}(1)}_{1}$

Q. $1 + 3 + 5 + 7 + \dots n$

```
~~void sum(n~~
int sum(int);
void main()
{
   int n;
```

```
    scanf("%d", &n);
    printf("%d", sum(n));
    getch();
  }
int sum(int x)
{
   if(x==1)
        return 1;
   else
        return(x+sum(x-2));
  }
```

Q. WAP to print table of a no. using recurssive f^n.

```
~~int to~~
    ~~table(int)~~;                    ~~table(int, int)~~;
~~void main()~~                        ~~void main()~~
{                                      ~~{~~
   ~~int n;~~                             ~~int n~~,
   ~~scanf("%d", &n)~~;                   ~~scanf("%d", &n)~~;
   ~~table(n)~~;                          ~~for(i=0; i<=10; i++)~~
   ~~getch()~~;                         ~~{~~
~~}~~                                     ~~printf("%d, tab(n, i)~~

     ~~table(int x)~~.
~~{~~
```

```
void table(int, int);          void table(int x, int i)
void main()                    {
 {                               if(i==11)
    int  n;                        return;
   scanf("%d", &n);              else
   tab(n, 1);                    { printf("%d", x*i);
   getch();                           table(x, i++);
 }                               }
```

Q. WAP to print fabunacci series for n positions

```
 int
~~void~~ fab(int);
void main()
{
    ~~int a=0, b=1, n~~
    int n, i;
    scanf("%d", &n);
    for(i=0; i<n; i++)
    {
        printf("%d", fab(i));
    }
    getch();
}
int fab(int p)
{
        if(p==0)
            return(0);
        else if(p==1)
            return(1);
        else
            return(fab(p-1)+fab(p-2));
}
```

12|04|17

Q. WAP to implement power function using f^n for +ve base & exponent.

```
int pow(int, int);
void main()
{
    int a, b
    scanf("%d %d", &a, &b);
        printf("%d", power(a,b));
    getch();
```

```
int pow(int x, int n)
{
    if (n == 0)          |   if (a == 1)
        return 1;        |       return (a);

    else
        return (a * pow(a * b-1);

}
```

STORAGE CLASSES

① Automatic Storage Class (RAM) — auto int a

② Register storage Class (Registers) — register int b

③ Static storage Class (RAM) — static int a

④ ~~Dynamic stor~~ External storage Class (RAM) - extern int b

By default initial values

auto - garbage		external - 0
register - garbage		
static - 0		

static variable can be initialize only a single time throughout the program.

Pain-u-lee Bus Service 17/04/17

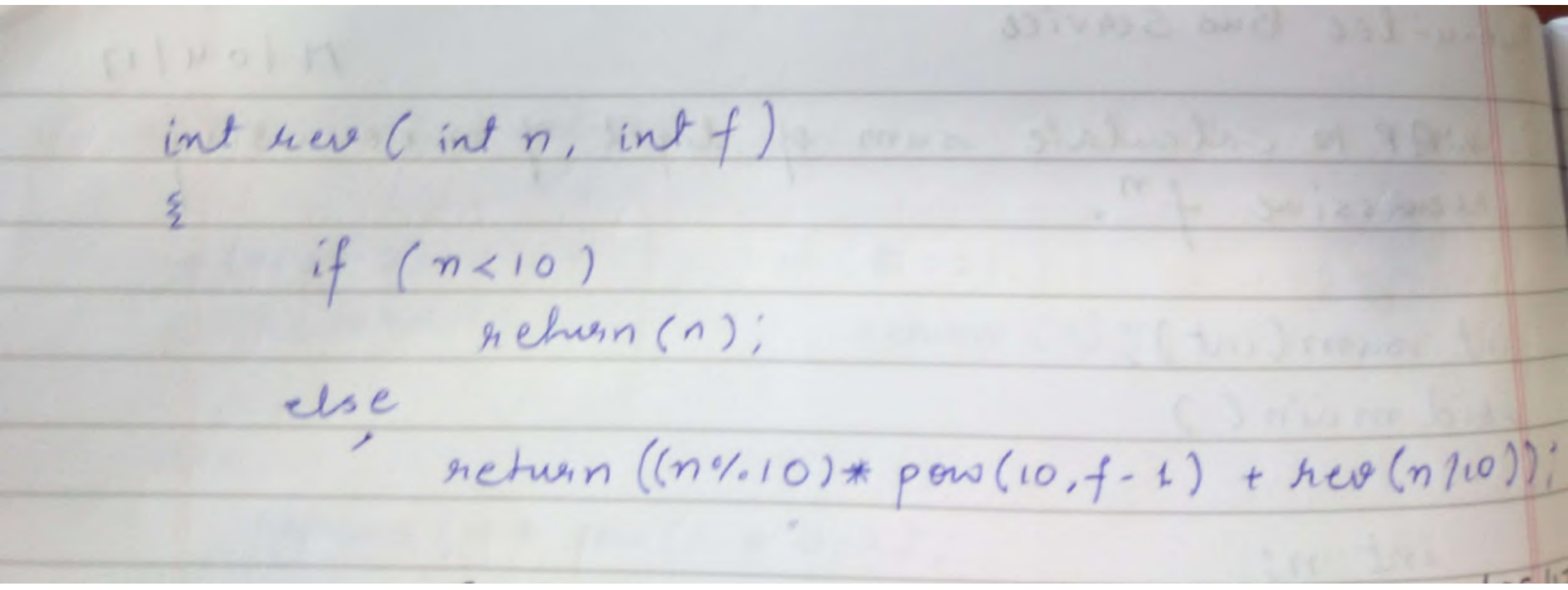
Q. WAP to calculate sum of digit of a no. using recurssive f^n.

```
int sum(int);
void main()
{
    int n;
    scanf("%d", &n);
    printf("%d", sum(n));
    getch();
}
int sum(int n)
{
    if(n<10)
        return n;
    else
        return(n %10 + sum(n/10));
}
```

Q. WAP to print reverse of a no. using recurssive f^n

~~int rev(int, int)~~
~~void main()~~
~~{~~
~~int n;~~ $\boxed{1}\boxed{2}\boxed{3}$
~~scanf("%d", &n);~~
~~printf("%d", rev(n));~~
~~getch();~~
~~}~~
~~int rev(int n)~~
{
~~if(n<10)~~
~~return n;~~
~~else~~
~~return n(~~
~~}~~

```
int rev(int, int);
void main()
{
    int n, flag=0;
    scanf("%d", &n);
    while(n>0)
    {
        n=n%10;
        flag++
    }
    printf("%d", rev(n, flag));
    getch();
}
```



```
int rev ( int n, int f)
{
    if (n<10)
        return(n);
    else
        return ((n%10)* pow(10,f-1) + rev(n/10));
```