

3
20/03/17

FUNCTIONS

function declaration :

return-type f-name(list of datatype of parameters)

int f (int, char);

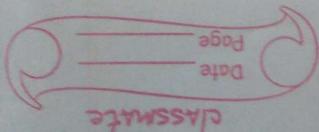
function definition :

return-type f-name (list of data type # parameters with name)

int f (int x, char ch)

function call :

[var =] f (var/const/exp);
(other than void)



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

```
int sum(int, int)
```

```
void main()
```

```
{
```

```
    int a, b, sum;
```

```
    sum = sum(a, b);
```

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

```
    sum = sum(a, b);
```

```
    printf("%d", sum);
```

```
}
```

```
    printf("%d", sum(a, b));
```

```
int sum(int x, int y)
```

```
{
```

```
    int temp;
```

```
    temp = x + y;
```

```
    return(x + y);
```

```
return(temp);
```

```
}
```

Q. WAP to calculate the following expression

~~float~~ fac(int); ~~int, int~~

```
void main()
```

```
{ float res;
```

```
int a, b, c, 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);
```

classmate

```
res = fa / (fb * fc);
```

```
printf("%d", res);
```

int
~~int~~ fac(int x) ~~int y + int z~~
~~x~~ $\leftarrow \begin{cases} & \\ & \end{cases}$ ~~y~~
~~x~~ $\leftarrow \begin{cases} & \\ & \end{cases}$ ~~z~~
 int f = 1, i;
 for (i = 2; i <= x; i++)
 $\leftarrow \begin{cases} & \\ & \end{cases}$ f = f * i;
 $\leftarrow \begin{cases} & \\ & \end{cases}$
 return(f);
 $\leftarrow \begin{cases} & \\ & \end{cases}$

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

int pal(int);

void main()

$\leftarrow \begin{cases} & \\ & \end{cases}$

int num, res
 scanf("%d", &num);
 res = pal(num);

if (res == 1)

$\leftarrow \begin{cases} & \\ & \end{cases}$ printf("Pal");

$\leftarrow \begin{cases} & \\ & \end{cases}$

else

$\leftarrow \begin{cases} & \\ & \end{cases}$ printf("NAP");

$\leftarrow \begin{cases} & \\ & \end{cases}$ getch();

$\leftarrow \begin{cases} & \\ & \end{cases}$

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);

}

}

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

27/03/17

```
int power(int, int)           pow()

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

int power(int x, int y)
{
    int i, p = 1;
    for(i=0, i<=y; i++)
    {
        p = p * x;
    }
    return(p);
}
```

Q. WAP to calculate the sum of following series :

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

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

```
int input()
int power(int, int)
void output(int)
```

103/104
Q3) Sold matrix)

Take int x, a, i,
~~so that sum of~~ that sum is 0;
x = input();
a = input();

int input(int n)
{
 int a;
 scanf("%d", &a);
 for (i=1; i<=n; i+=2)
 {
 Sum += power(x, i);
 }
}

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

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

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

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 argument and return type.

- ① No return type No arguments
- ② No return type with arguments
- ③ With return type but no arguments.
- ④ With return type with arguments
- 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(int x, int y)
{
 return(x+y);
}

Q. WAP to calculate the following expⁿ

$$nC_a = \frac{n!}{a!(n-a)!}$$

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

{

calc();

getch();

{

void calc();

{ int n, a;

res = ~~fact(n)~~ (fact(n)/(fact(n)*fact(n-a)));
output(res);

{

int fac(int x)

{ int n, a, f=1, i;

n = input();

a = input();

for(i=2; i <= x; i++)

{ f = f * i

{

return(f);

{

int input()

{ int x

scanf("%d", &x);

{

void output(float x)

{ printf("%.4f", x);

{

```

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 = fact(n)/(fact(r)*fact(n-r));
    output(res);
}

int input()
{
    int x;
    scanf("%d", &x);
    return(x);
}

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

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

(1) void calculate(int, int)
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.

<code>(int [], int)</code>	<code> </code>	<code>(int [], int, int)</code>
↳ size		

Return type f-name (datatype with []);

f-name (name of array);

Return type f-name (datatype with parameters [])

{

}

`void f(float []);`

`void main()`

{

`float a[5];`

`f(a);`

}

→ call by reference

Scanned by CamScanner

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

Return type f-name (datatype *);

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

Return type 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 ~~no.~~ 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[i] = a[j];  
a[j] = temp;
```

{

```
for(i=0; i<n; i++)
```

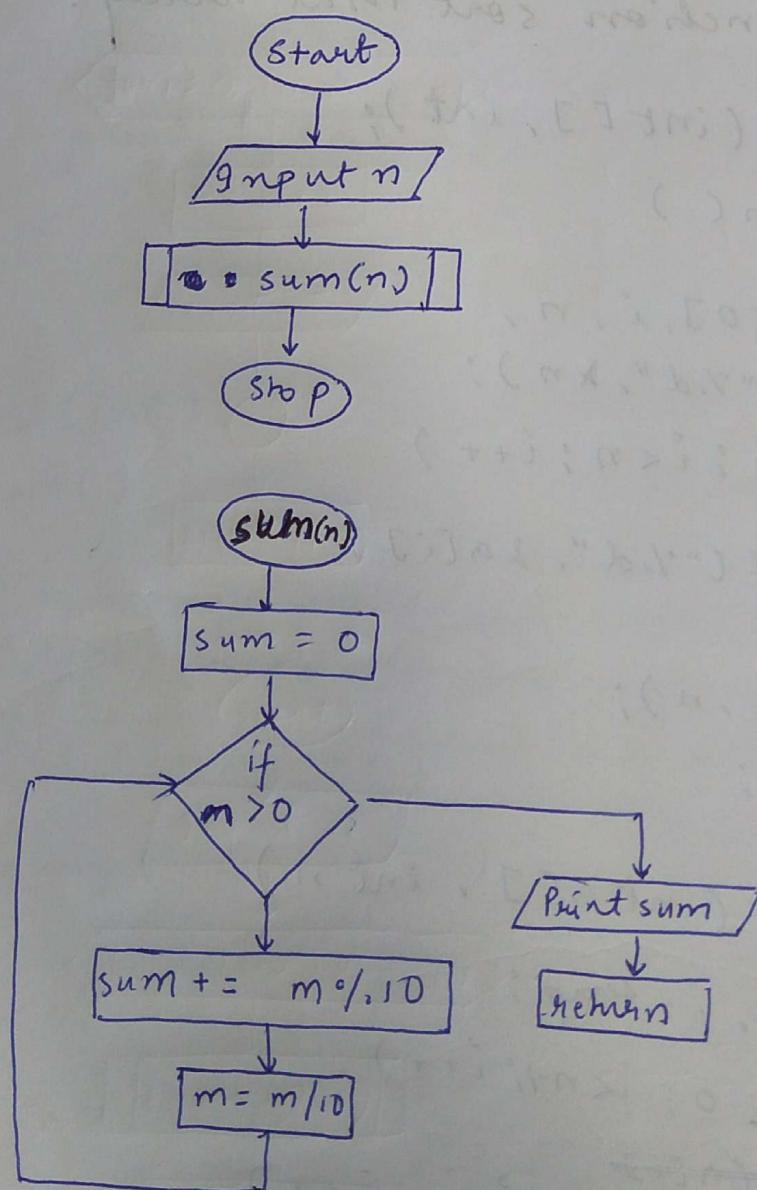
{

```
    printf ("%d", &a[i]);
```

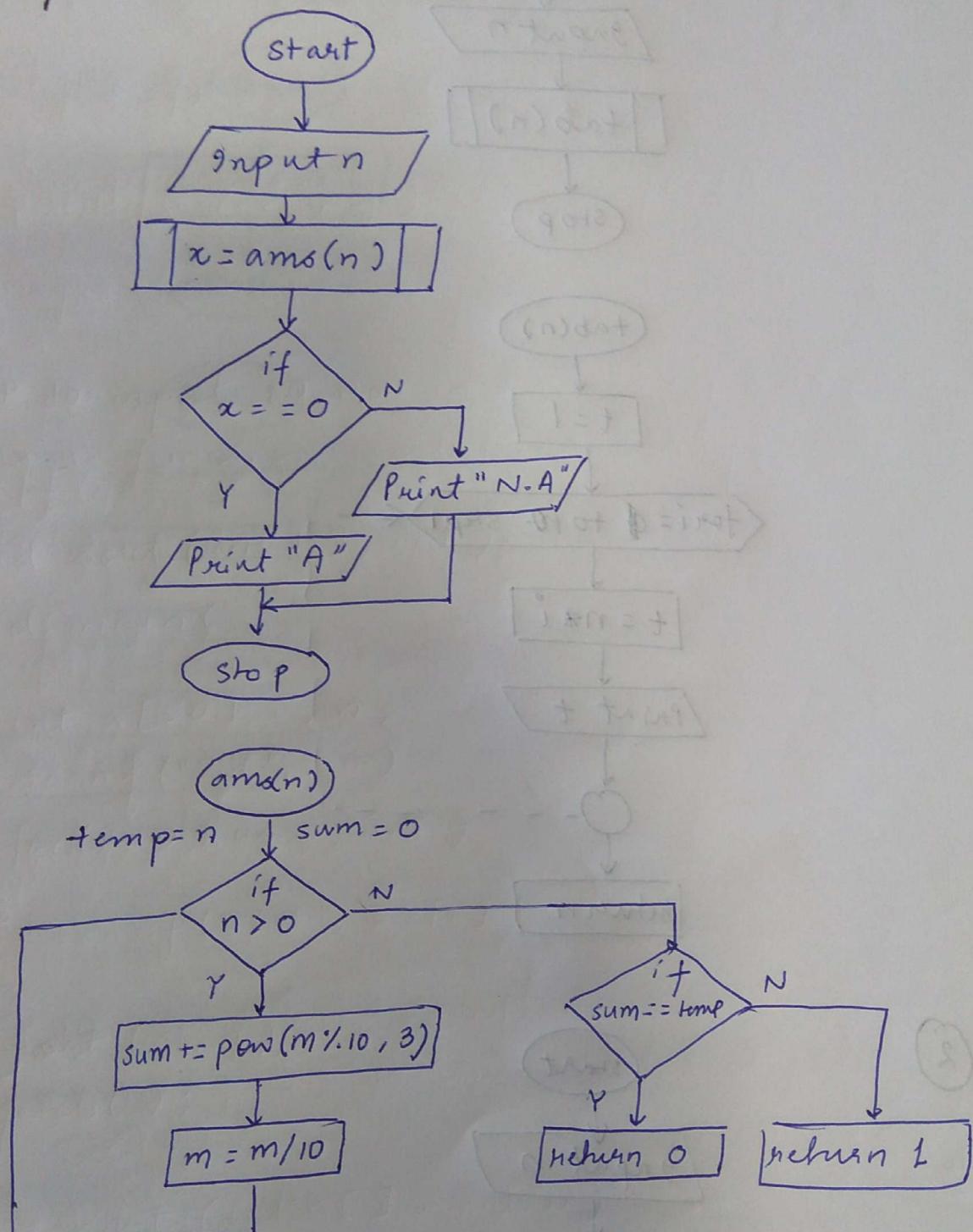
{

{

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

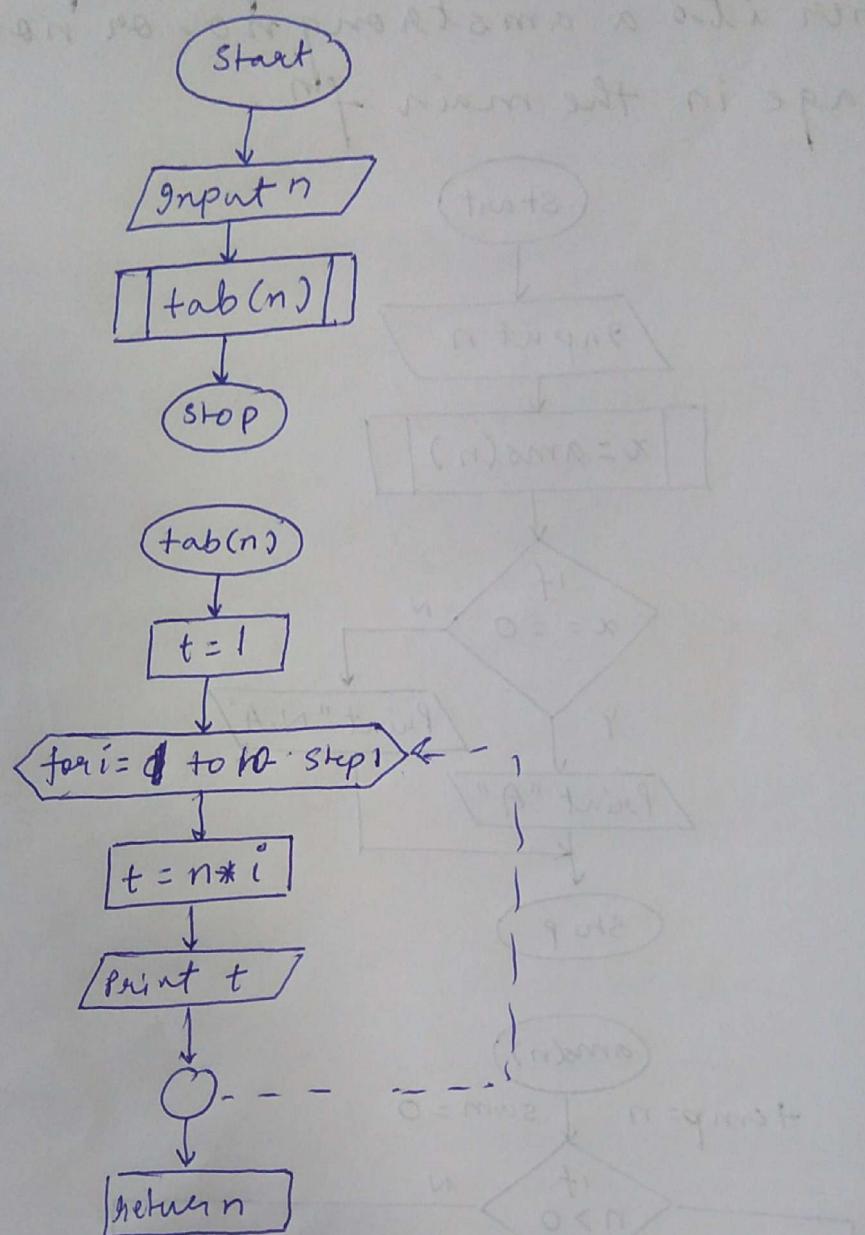


Q) AF to input a no. and by using af^n check whether its a amstrong no. or not. Print the message in the main fn.

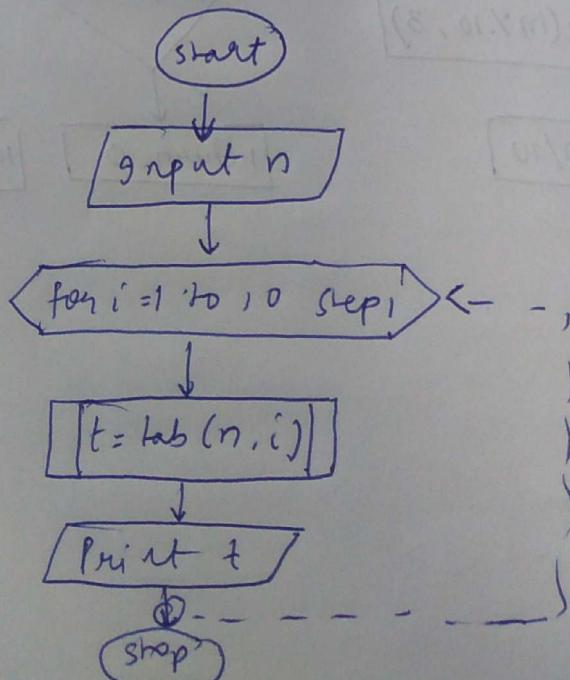


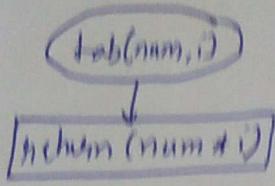
Ques. to input a no. and print its table using fⁿ

①

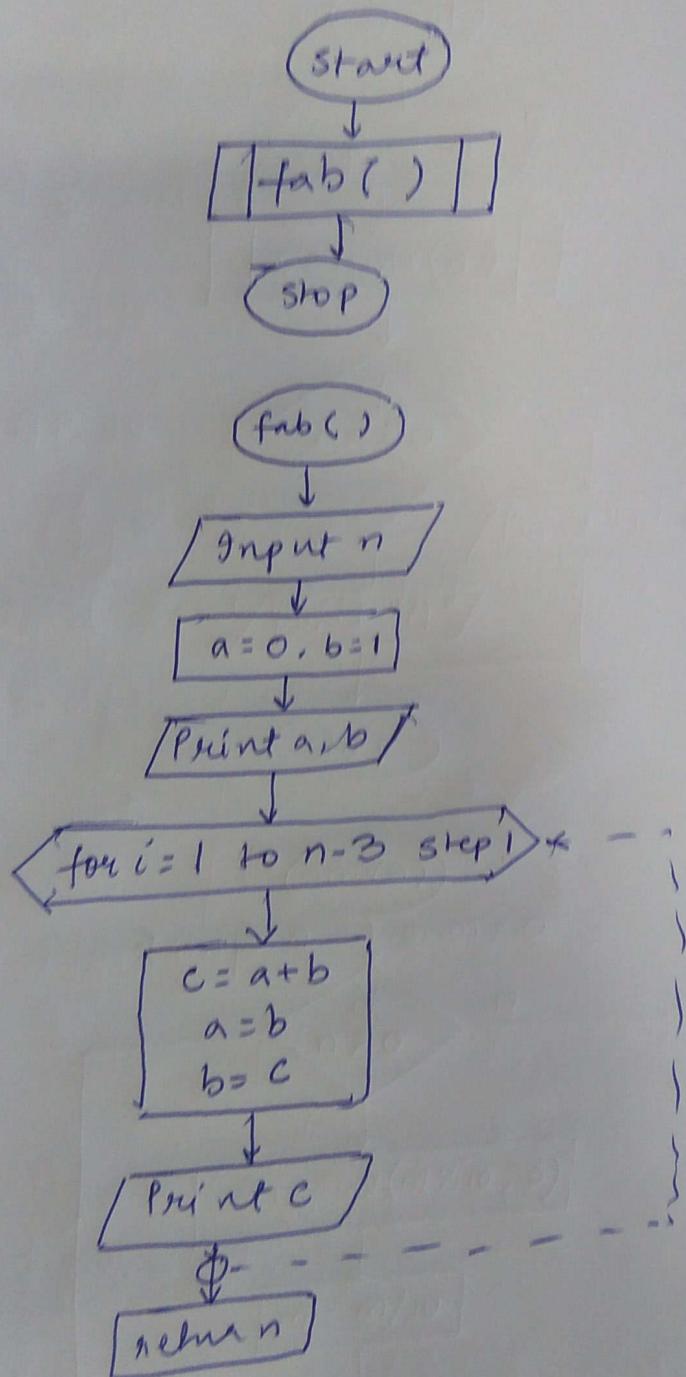


②





DAF to print fabunacci series using function.



Recursion

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

Q. WAP to calculate factorial of a no. using recursion 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

$$1 + 2 + 3 + \dots + n$$

```
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 ($x + \text{sum}(x-1)$);

}

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));

}

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

~~void sum()~~

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 - 1));
}

```

Q. WAP to print table of a no. using recursive fn.

int to

```

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

```

table (int x).

```

void table (int, int);
void main()
{
    int n;
    scanf ("%d", &n);
    tab (n, 1);
    getch();
}

```

~~table (int, int);~~

~~void main()~~

```

int n,
scanf ("%d", &n);
for (i = 1; i <= 10; i++)
{
    printf ("%d", tab (n, i));
}

```

~~void table (int x, int i)~~

```

if (i == 11)
    return;
else
    {
        printf ("%d", x * i);
        table (x, i + 1);
    }
}

```

Q. WAP to print fabunacci series for n positions

```
int  
void fab(int);  
void main()  
{  
    int a=0, b=1, n;  
    int 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));  
}
```

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)
        return 1;
    else
        return (x * pow(x, n - 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 storage class (RAM) - extern int b

By default initial values

auto - garbage

extern - 0

register - garbage

static - 0

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

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

```

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

int sum(int n)
{
    if (n <= 0)
        return 0;
    else
        return (n % 10 + sum(n / 10));
}

```

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

```

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

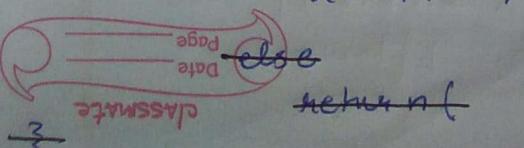
int rev(int n)
{
    if (n < 10)
        return n;
    else
        return n % 10 + 10 * rev(n / 10);
}

```

```

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));