6/9/18

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

$$= \frac{FACT(n)}{FACT(r)\,F(n-r)}$$

#define FACT

Implement $(1+x)^n$.

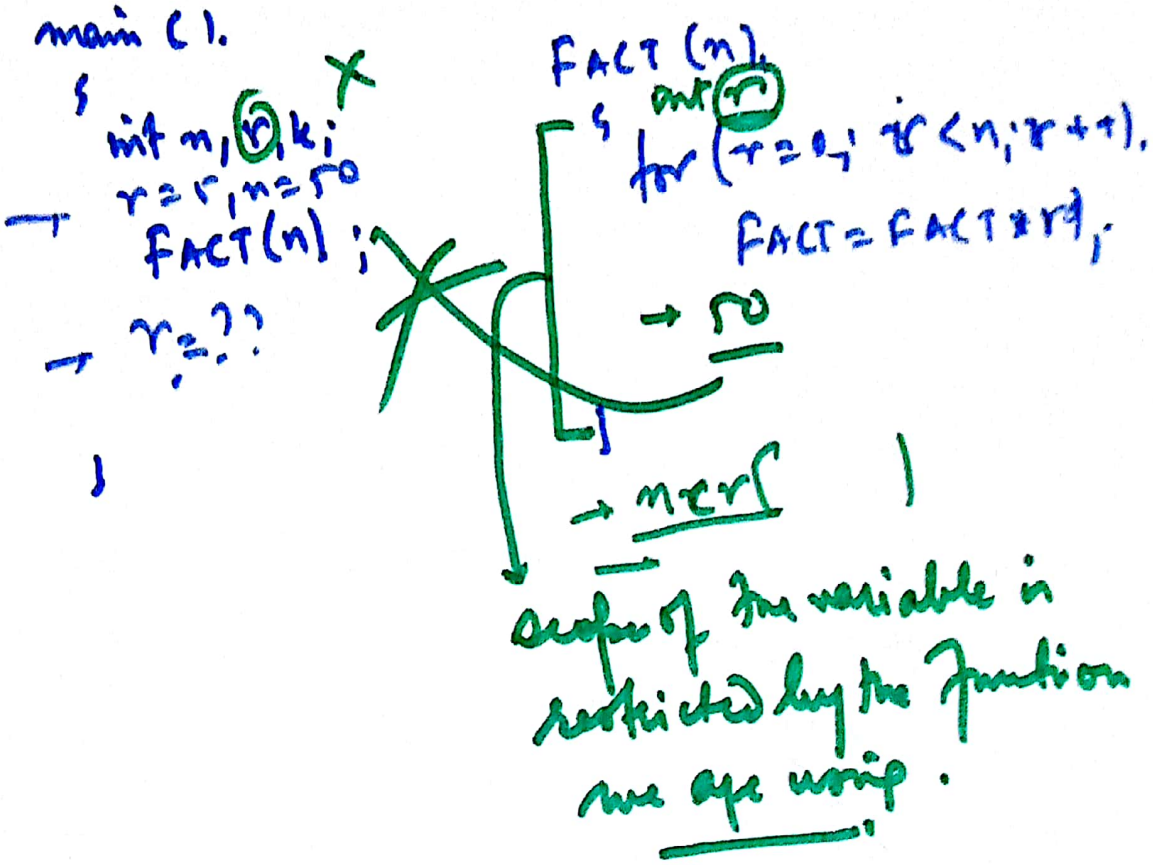
$$= \binom{n}{0}x^0 + \binom{n}{1}x^1 \cdots + \binom{n}{n}x^n$$

#include<stdio.h> int ncr(int n, int r).

int x^k = pow(x,k).

main().
{ scanf("%d%d", &x, &n); sum = 0
for(i=0; i<=n; i++)
{
sum += ncr(n,i) * pow(x,i);
}
}

main().
{
k = FACT(n);
}




```
int ncr(int n).
{ int r;=0;
  return(n!/(r!*(n-r)!));
}
```

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

main( )
 { r=0;
  for (i=0; i <= n; i++r)
    {
    sum+ = ncr(n),* pow(x,i).
     r++j

     }
  }

ncr (int n).
 { int r ?!
  return (n! FACT(n)/(FACT(r)*
                      FACT(n-r)).
 }
```

→ Now we know a way to transfer value from function to function.

——→ Call by value

```
int
float totalF(int B[]).
{  int total;
   for(i=0, i<ASIZE; i++)
   {
      total = total + B[i];
      B[i] = B[i] - i;
   }
}
```

```
main().
{ int total M
  int x = {1, 5, 7, 6, 2}

  totalF = totalF(&x);
  for(i=0; i<ASIZE; i++).
    printf("%d", x[i]);
```

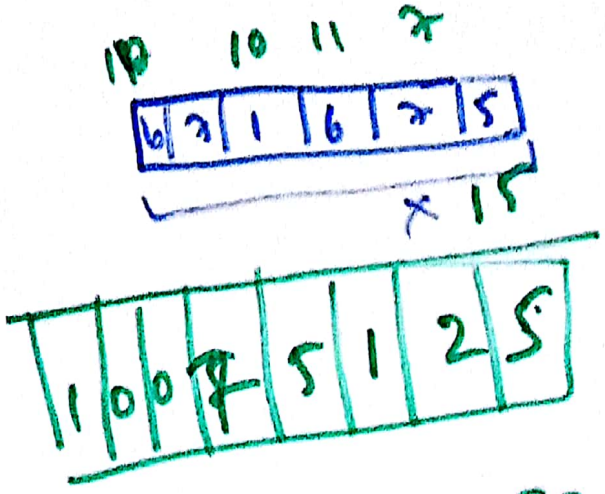
address of the first element

x[0] → x
x[1] → x+1

1 5 7 6 2

→ 0, 4, 6, 5, 1

→ 1, 4, 5, 3, -2.




```
int multiply_big_no(int a[], int x, int length)
{
}

int multiply_big_no(int a[], int x, int length)
{
return (length_new);
}

main().
for(i=0; i<x; i++)

100/100

int large_fact(int a[], int x)
{ a[0]=1; length=1;
for(i=1; i<=n; i++)
length = multiply_big_no(a, x, length);
/* FACT = FACT * i */
return length.
}
```