

Recursion :-

- you use a function
- Example : using Recursion to find The factorial

```
int fact (int i)
{
    if (i <= 1)
        return 1;
    else
        return i * fact (i-1);
}
```

Programs to do using Recursion :-

- × Reversing strings
- Multiplying ✓
- dividing ✓
- fibonacci (by heart) ✓
- greatest common divider page (535)

Alam Ehsani

• what's the output:-

(base)^{power}

```

1- int
   power_raiser ( int base , int power)
   {
       int ans;
       int base = 2 , power = 3;
       if (power == 0)
           ans = 1;
       else
           ans = ( base * power_raiser (base, power-1)
   }
   return ans;
}

```

Answer is 100

```

int fun (a[], int n)
{
    int x;
    if (n == 1)
        return a[0];
    else
        x = fun (a, n-1);
    if (x > a[n-1])
        return x;
    else
        return a[n-1];
}

```

ans = 8

```

2- int strange (int n);
    ;
    ;

```

```

n=7
int strange (int n)
{
    n = 7;
    int ans;
    if (n == 1)
        ans = 0;
    else
        ans = 1 + strange (n/2);
}
return ans;
}

```

n=5 →
 • Assuming n=5
 a[] = {12, 10, 30, 50, 100}

ans = 2
 if n = 9
 ans = 3

Alaa Etaiwi

work on y

5- write a program of multiplying using recursion

```

void find_mult (int, int);

```

```

int main ()
{
    int x; int y; ans;

```

find
 $x \times y$

Pr -
Sc -

```

ans = find_mult (x, y);

```

```

}
int find_mult (int x, int y)
{

```

```

    if (y == 0)
        return 0;

```

```

    else
        return (x + find_mult (x, y-1));

```

2, 3
 $2 + (2, 2)$
 $2 + 2 + 2 + (2, 1)$
 2, 3
 $2 + (2, 2)$
 $2 + 2 + (2, 1)$
 $2 + 2 + 2 + (2, 1) = 6$

(9, 3)
 $9 - 3$
 $6 - 3$
 $3 - 3 = 0$
 $9 - 6 = 3$
 $5 + 6 = 10 - 5$
 $10 - 5 = 5 - 5 = 0$
 (9, 3)

6- Division :- You change the function to :-

```

    if (x == 0)
        return 0;

```

```

    else
        return 1 + find_mult (x-y, y);

```

work on x

7- greatest gcd (int m, int n)

greatest common divider

```

    int ans;
    if (m % n == 0)
        ans = n;

```

```

    else
        ans = gcd (n, m % n) -> return (ans)

```

16 3
 3 1
 16 3 2
 3 2 16
 $1 + (6, 3)$
 (3, 3)

• Write a Program that :-

1- Calculate the sum of successive integers starting at 1 \rightarrow n

```
int find_sum(int n)
{
    int ans;
    if (n == 0)
        ans = 0;
    else
        ans = n + find_sum(n-1);
    return ans;
}
```

```
void fun (int x)
{
    if (x > 0)
    {
        fun(--x);
        printf("%d\n", x);
        fun(--x);
    }
}
x = 4
output: - 0 1 2 0 3 0 1
```

??
2-

compute a pair of fibonacci numbers (

~~X~~ $F(n+1), F(n)$ (one recursive call)
|
n

```
int fast_fib (int n, int n+1)
{
    if (n == 1)
        return 0;
}
```

Alan Bhaiwi

3- A program that sorts n elements of an array of integers

```
void BubbleSort (int arr[], int n)
```

1 | 4 | 2 | 0 | 3

1 2 3
1 2 4 4 4

```
if (n == 1)
    return
```

if (n >= 1) set array[0] = 1

```
printf ("n=1 is true.");
```

select (4, 2)

```
else
```

```
{ for (i = 0; i < n; i++)
    if (arr[i] > arr[i+1])
        temp = arr[i+1];
        arr[i+1] = arr[i];
        arr[i] = temp;
    BubbleSort (arr, n-1)
}
```

```
int fun (int a, int b)
```

```
{ if (b == 0)
    return 0;
  if (b % 2 == 0)
    return fun (a+a, b/2);
```

```
return fun (a+a, b/2)
      + a;
```

fun(4, 3) → answer = 12

4- fibonacci:-

1 1 2 3 5 8 13
n = 1 2 3 4 5 6 7

```
if (n == 1 || n == 2)
    return 1;
```

```
else
```

```
return (fib(n-1) + fib(n-2));
```

3 2
2 1
1 + 1 + 1



Ataa Etaiwi