



الأُسبوع "الحادي عشر" برمجة حاسوب "C++" اسم الطالبة: سندس عوده اسم الدكتور: د. محمد خليل

Recursion

بالمعنى البسيط هو استدعاء التمجس لنفسه لكن بموايد
توسيط نفس انهاء عملية الريبورفين ببيس كين معينة

```
void message (int times)
```

```
{
```

```
if (times > 0) {
```

```
cout << "This is a recursive function. \n";
```

```
message (times - 1);
```

```
}
```

```
return;
```

```
}
```

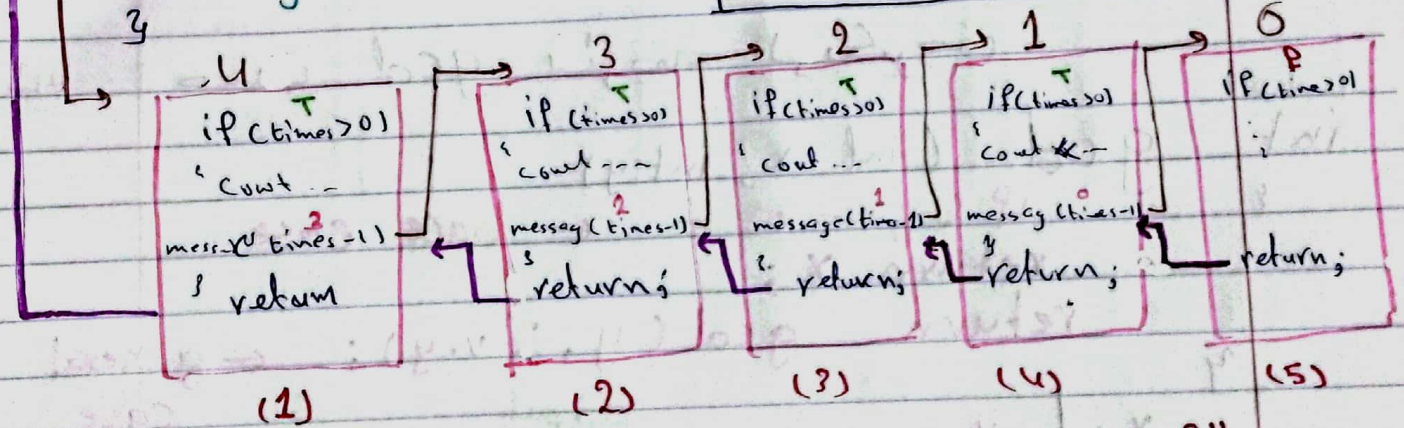
```
int main ()
```

```
{
```

```
message (4);
```

```
}
```

This is a recursive function
This is a recursive function
This is a recursive function
This is a recursive function



الشرح :-

في (1) تم استدعاء التمجس من main وارحال قلية للبرامير times
تساوي 4 وتم سأل هل $4 > 0$ فكان الجواب T قام بطباعة
الجملة مع التامة وتم قام باستدعاء نفسه في الطر الثالث داخل التمجس
ولكن كانت القيمة المرسلة هي عبارة عن $(times - 1)$ اي (3 و 1 و 4)

وفي كل مرة حتى وصلنا الى التامة رقم (5) كذلك $(times = 0)$ في هذه الحالة ار ip

لا يدخل ال ip ويدرج الى return ليعود الى المرحلة السابقة (مكان الاستدعاء)
يسير في انهاء الاستدعاءات حتى يصل الى التامة الأخيرة ويعود الى مكان الاستدعاء
في ال main (1) الأولى

Writing Recursive Functions

① نحتاج إلى تعريف واضح للمشكلة التي نريد معالجتها عن طريق الـ **الريكرشن**.

② المشكلة التي يتم معالجتها في الـ **الريكرشن** تكون مشكلة كبيرة وينحاول في كل مرة أنه يأخذ المشكلة الأصغر.

③ لازم نحدد (base case) هي الحالة التي تنهي عمل الـ **الريكرشن** ما دام يكون في استبعاد للفتنجن (الريكرشن) عند الوصول لها.

④ لازم نحدد (general case) الحالة التي تعتمد على استبعاد الفتنجن الخاصة. لكن لازم يكون في كل مرة هذا الاستبعاد يقربني من (base case) ←

حساب gcd بطريقة الـ **الريكرشن**

```
int gcd (int x, int y)
{
    if (y == 0) ← base case
        return x;
    return gcd (y, x % y); ← general case
}
```

x	y
12	15
15	12
12	3
3	0

output

3

يمكن استخدام helper function

```
* int gcdRec ( )  
{  
    return gcdRec ( 75, 25 );  
}
```

```
* int gcdRec (int x, int y)  
{  
    if (y == 0)  
        return x;  
    return gcdRec (y, x % y);  
}
```

التجنيس overloading

في حال تم استدعاء الـ gcdRec من دون ارجاع x, y
يذهب الى الخطأ القيم الموضوعة في الـ helper function


```

int sum (int n)
{
    if (n <= 1)
        return n;
    return n + sum(n-1);
}

```

```

int main()
{

```

```

    int n = 4;

```

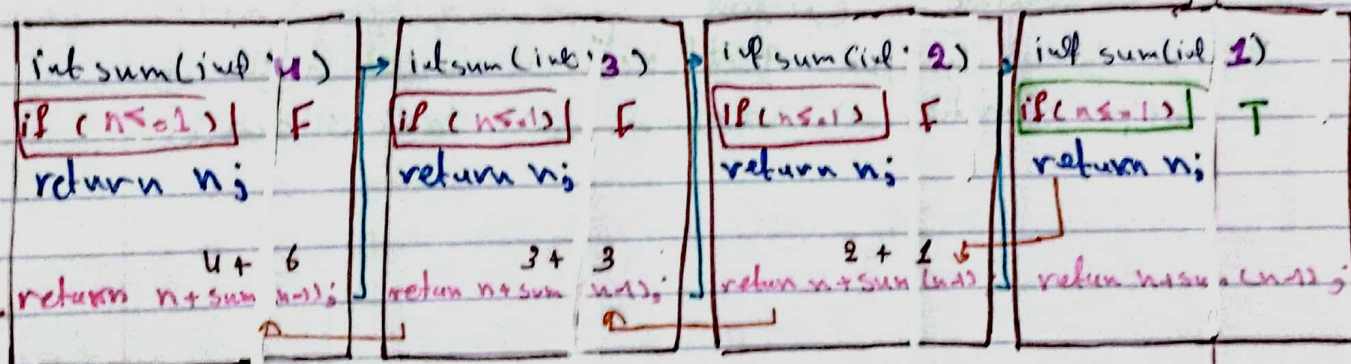
```

    cout << sum(4);
}

```

output:

10



Factorial

n n! $n \times (n-1)!$

5	5 * 4 * 3 * 2 * 1	5 * 4!	120
4	4 * 3 * 2 * 1	4 * 3!	24
3	3 * 2 * 1	3 * 2!	6
2	2 * 1	2 * 1!	2
1	1	1	1

```
int fact (int x)
{
```

```
    if (x <= 1)
```

```
        return x;
```

```
    return x * fact (x-1);
```

```
}
```

output

24

```
int main ()
```

```
{
```

```
    cout << fact (4);
```

24

3

```
int fact (int 4)
```

```
{ if (x <= 1)
```

```
    return x;
```

```
    return (x * fact (x-1));
```

4 * 6

```
int fact (int 3)
```

```
{ if (x <= 1)
```

```
    return x;
```

```
    return (x * fact (x-1));
```

3 * 2

```
int fact (int 2)
```

```
{ if (x <= 1)
```

```
    return x;
```

```
    return (x * fact (x-1));
```

2 * 1

```
int fact (int 1)
```

```
{ if (x <= 1)
```

```
    return x;
```

```
    return (x * fact (x-1));
```