



บทที่ 1 Introduction ทบทวนการโปรแกรม

1. ทบทวนการใช้งาน loop
2. Array
3. ทบทวน structure
4. เรียนการใช้งาน Pointer
5. Recursive



1) โครงสร้างภาษา :

```
1. #include <iostream>  
2. using namespace std;
```

Header

```
3. int main()  
4. { cout << "Hello World!" << endl;  
5. }
```

**Function
Main**



03603212 : Module1 – Introduction 3

```
for( i=1; i<=5; i++)  
    cout << "hello";
```

 5

```
for( i=0; i<=5; i++)  
    cout << "hello";
```

 6

```
for( i=1; i<5; i++)  
    cout << "hello";
```

 4

```
for( i=0; i<5; i++)  
    cout << "hello";
```

 5

```
for( i=-4; i<=4; i++)  
    cout << "hello";
```

 9

```
for( i=7; i<10; i++)  
    cout << "hello";
```

 3

```
for( i=1; i<=5; i=i+2)  
    cout << "hello";
```

 3

```
n=1;  
for( i=1; i<=n; i++)  
    cout << "hello";
```

 1



Loop

12345

12345

12345

12345

12345

12345

```
for( i=1; i<=5; i++)  
    cout << i;  
  
cout << endl;
```


```
for( i=1; i<=5; i++)  
{  
    for(j=1; j<=5; j++)  
        cout << "j";  
    cout << endl;  
}
```



1 12 123 1234

```
for(j=1;j<=1;j++)  
    cout << j;  
for(j=1;j<=2;j++)  
    cout << j;  
for(j=1;j<=3;j++)  
    cout << j;  
for(j=1;j<=4;j++)  
    cout << j;
```

```
for(i=1;i<=4;i++)  
{   for(j=1.....;j<=i.....;j++)  
        cout << j;  
    cout << " ";  
}
```





Loop

1

12

123

1234

12345

```
for( i=1; i<=5; i++)  
{   for(j=1; j<=i; j++)  
        cout << "j";  
        cout << endl;  
}
```



การบ้าน

1.รับ input เป็นจำนวนเต็ม 1 ค่า ใช้ loop พิมพ์ตัวเลขดังด้านล่าง

input : 4

1 12 123 1234

2 23 234

3 34

4

4 43 432 4321

3 32 321

2 21

1



การบ้าน

2. รับ input เป็นจำนวนเต็ม 1 ค่า แสดงว่าค่านั้นเป็นจำนวนเฉพาะหรือไม่

Hint : ใช้การ modulus %

input : 4

Not prime

หรือ

input : 11

Prime

```
for (i=1; i<=4; i++) {  
    if (n % i == 0)  
        cout << "Not prime";  
}
```

```
for (i=1; i<=11; i++) {  
    if (n % i != 0)  
        cout << "prime";  
}
```




โปรแกรมสลับค่า

```
#include <iostream>
using namespace std;
int main()
{
    int a=1,b=2,tmp;
    cout << a <<" "<< b << endl;
    tmp=a; 1
    a=b; 2
    b=tmp; 1
    cout << a <<" "<< b << endl;
}
```

```
int a=1 ; b= 2, tmp
tmp= b; 2
b = a; 1
a = tmp; 2
cout << a <<" "<< b << endl;
```



a = 1
b = 2

x = a;
y = b;
a = y;
b = x;

ค่าในกล่องเก่า
↑
จะค่าเปลี่ยนที่ค่าเดิม



2)Array

2.1) Array 1 มิติ

```
int a[5];  
int score[5]={20,15,17,13,8};
```

score	20	15	17	13	8
0	1	2	3	4	

`tmp = score[0];`

`score[0] = score[4];`

`score[4] = tmp;`

```
int main()  
{  
    int score[5]={20,15,17,13,8};  
  
    for(int i=0;i<=4;i++)  
        cout << score[i] << " ";  
}
```



โปรแกรมสลับค่า

0	1	2	3		4
10	20	30	40	...	100

0	1	2	3		4
100	20	30	40	...	10

```
#include <iostream>
using namespace std;
int main()
{ int tmp;
  int score[5]={10,20,30,40,
    50,60,70,80,90,100};
  for(int i=0;i<=4;i++)
    cout << score[i] << " ";
  cout << endl;
```

```
    tmp=score[0];
    score[0]=score[4];
    score[4]=tmp;
    for(int i=0;i<=4;i++)
      cout << score[i] << " ";
    cout << endl;
  }
```



การบ้าน

3. จงใช้ loop เขียนโปรแกรมสลับค่าตัวเลขในอะเรย์กลับด้าน จาก 10-100 ให้มีค่าเป็น 100-10

หมายเหตุ กำหนดให้ใช้ loop ห้ามใช้การสลับค่าด้วย index ตรงๆ

```
tmp=score[0];  
score[0]=score[9];  
score[9]=tmp;
```

} ห้าม

0	1	2	3	...	9
10	20	30	40	...	100

++i

++j

0	1	2	3	...	9
100	90	80	70	...	10



2.2 Array 2 มิติ

int ชื่อตัวแปร [แถว][หลัก];

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

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{    char a[3][3]={ {'a','b','c'},  
                    {'d','e','f'},  
                    {'g','h','i'}};
```

```
    int i,j;
```

```
    for(i=0;i<3;i++)
```

```
    {    for(j=0;j<3;j++)
```

```
        cout << a[i][j] << " ";
```

```
        cout << endl;
```

```
    }
```

```
}
```

j = 0

j = 0, 1, 2

i = 1

j = 0, 1, 2

i = 2

j = 0, 1, 2



2.3 Sample word puzzle

	1	2	3	4
1	T	H	I	S
2	W	A	T	S
3	O	A	H	G
4	F	G	D	T

THIS
HIS
IS
FAT
THAT
HI
IT
TWO
HAT
AT



```
#include <Iostream>
using namespace std;
int main()
{
    int i,j,k;
    char a[4][4]={
        {'T','H','I','S'},
        {'W','A','T','S'},
        {'O','A','H','G'},
        {'F','G','D','T'}};

    string db[12] = {"IS", "THIS", "HIS", "AT", "HI",
                    "IT", "TWO", "OF", "FAT", "THAT", "HAT"};
    string temp="";
    for(i=0;i<4;i++)
    {
        for(j=i;j<4;j++)
        {
            for(k=i;k<=j;k++)
            {
                temp=temp+a[0][k];
            }
            cout << temp;
            getchar();
            temp.clear();
        }
        cout << endl;
    }
}
```

T TH THI THIS
H HI HIS
I IS
S



```
for(i=0;i<4;i++)
{
    for(j=i;j<4;j++)
    {
        for(k=i;k<=j;k++)
        {
            temp=temp+a[0][k];
        }
        cout << temp;

        cout << " " << temp.size(); //อยากรู้ขนาด

        if(temp==db[0])
            cout << "*** Yes***\n";

        temp.clear(); //clear คำสั่ง
    }
    cout << endl;
}
```

T TH THI THIS
H HI HIS
I IS
S

วิธีการเทียบ
คิดอย่างไร

db

IS
THIS
HIS
AT
HI
IT
TWO
OF
FAT
THAT
HAT



การบ้าน

4. กำหนดตาราง puzzle ดังภาพให้นิสิตเขียนโปรแกรมทำการค้นหา
คำในตารางว่ามีคำที่อยู่ในดิกชันนารีกี่คำ

กำหนดคำในดิกชันนารีดังนี้

"IS", "THIS", "HIS", "AT", "YOU", "HI", "IT", "TWO",
"OF", "FAT", "THAT", "HAT", "GOD", "CAT",
"HAT", "AN", "FOUR"

	1	2	3	4
1	T	H	I	S
2	W	A	T	S
3	O	A	H	G
4	F	G	D	T

Output

T
TH
THI
THIS Found 1
H
HI Found 2
..
Total =15

//คะแนนขึ้นกับความสามารถที่เจอค่ะ ถูก 5-10 ตัว
75% ถูกเกิน 10 ตัว 100% ถ้ามีตัวผิดพลาดด้วย
เหลือ 50%



3. Structure

Structure is a user-defined datatype which allows us to combine data of different types together.

```
struct student
```

```
{  int Id;
```

```
    float Grade;
```

```
    char Gender;
```

```
};
```

```
struct student std1;
```

std1

int	Id:	
float	Grade:	
char	Gender:	

การใช้งาน

1. Definition structure
2. ประกาศตัวแปร
3. การใส่ค่าลงในตัวแปร
4. การดึงค่ามาใช้



```
struct student  
{ 4 int Id;  
  4 float Grade;  
  1 char Gender;  
}std2; g
```

std2

int	Id:	
float	Grade:	
char	Gender:	



3.1 การ defined structure และประกาศตัวแปร

```
struct employee
```

```
{  int Id;  
    string Name;  
    float Salary;  
};
```

```
struct employee member1, member2;
```

member1

int	Id:	
string	Name:	
float	Salary:	

member2

int	Id:	
string	Name:	
float	Salary:	



3.2 การใส่ค่าลงในตัวแปร structure

```
member1.Id=100;  
member1.Name= "Somchai";  
member1.Salary=40000;
```

101
Jane
41000

```
cin >> member2.Id;  
cin >> member2.Name;  
cin >> member2.Salary;
```

member1

int	Id:	
char[]	Name:	
float	Salary:	

member2

int	Id:	
char[]	Name:	
float	Salary:	



3.3 structure initialization(ใส่ค่าเริ่มต้น)

```
struct structname variable = { val1, val2, ... };
```

member3

int	Id:	102
string	Name:	Kuer
float	Salary:	50000

```
struct employee member3 = { 102, "Kuer", 50000};
```

type

ค่าเริ่มต้น

4. Pointer ตัวชี้ตำแหน่ง (เก็บพ้องงานกับจำนวนเลข ที่)

1. Declaration of a pointer

```
int x, *ptr;
```

2. Pointer Operator

& Address Operator
* Indirection Operator

มีตัวชี้ตำแหน่งตัวอื่น

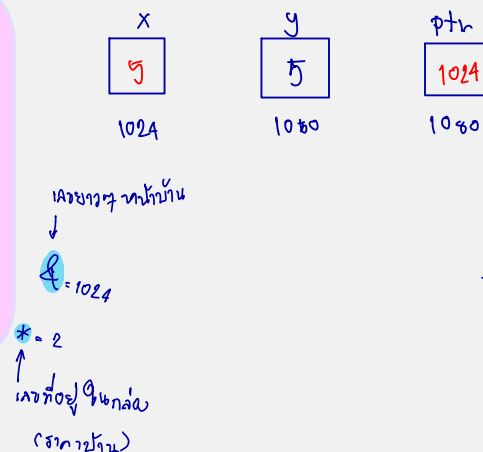
```
int x, y, *ptr;
```

```
x=5;
```

```
ptr = &x; 1024
```

```
y = *ptr; 5
```

y ↓ ชี้ไปเก็บ 1024

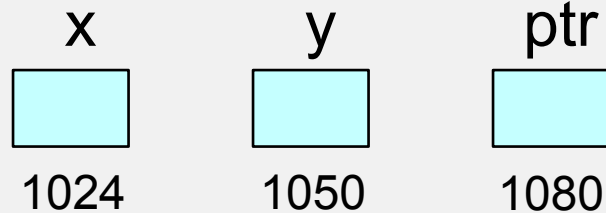


* คือตัวชี้ไปเก็บค่าเลขที่เก็บค่าตัวแปร

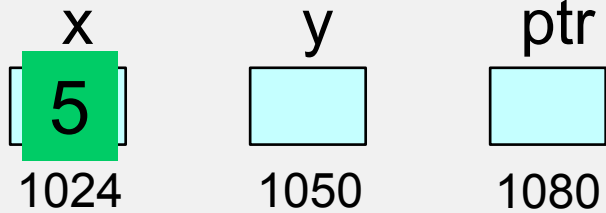


ตัวอย่าง 4.1

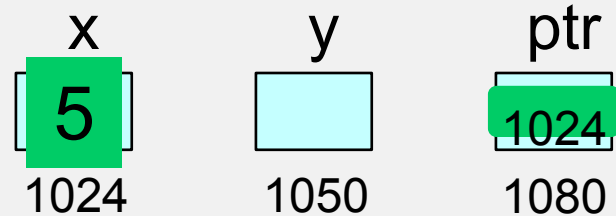
1. `int x, y, *ptr;`



2. `x=5;`

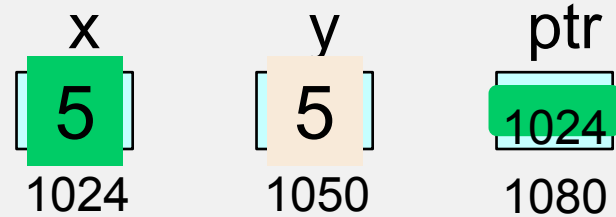


3. `ptr = &x;`

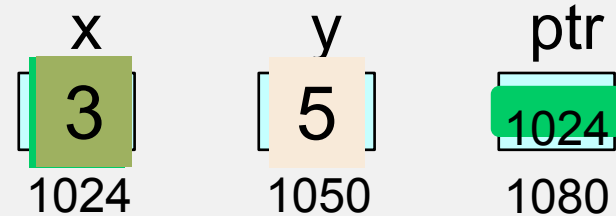




4. $y = *ptr;$



5. $*ptr = 3;$

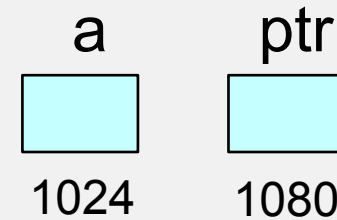


พอฉันเริ่ม 1024
ก็เก็บของแรก
*ptr คือ 5 ใน 1024
พอ *ptr เปลี่ยนเป็น 3
แล้ว ptr เก็บค่าคือ 1024
คือ 1024 แล้วมันเปลี่ยน
ตามที่เราเก็บค่าแรก



Exercise 1

- ประกาศตัวแปร ชื่อ a มี type float และประกาศตัวแปรชื่อ ptr มี type pointer to float กำหนดให้ a มีค่า 3.14



- กำหนดให้ ptr เก็บตำแหน่งของ a
- พิมพ์ค่า 3.14 โดยผ่านทาง ptr



ตัวอย่าง 4.2

```
1. #include <iostream>
2. using namespace std;
3. int main()
4. {   int a=10 ,*ptr;
5.     ptr = &a;
6.     cout << "*ptr      = " << *ptr << endl;
7.     cout << "a      = " << a << endl;
8. }
```

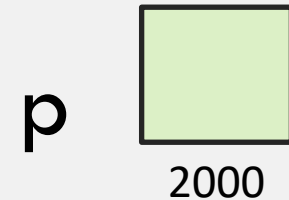
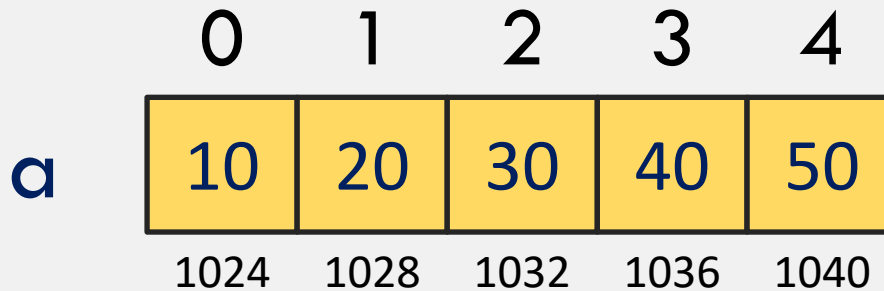


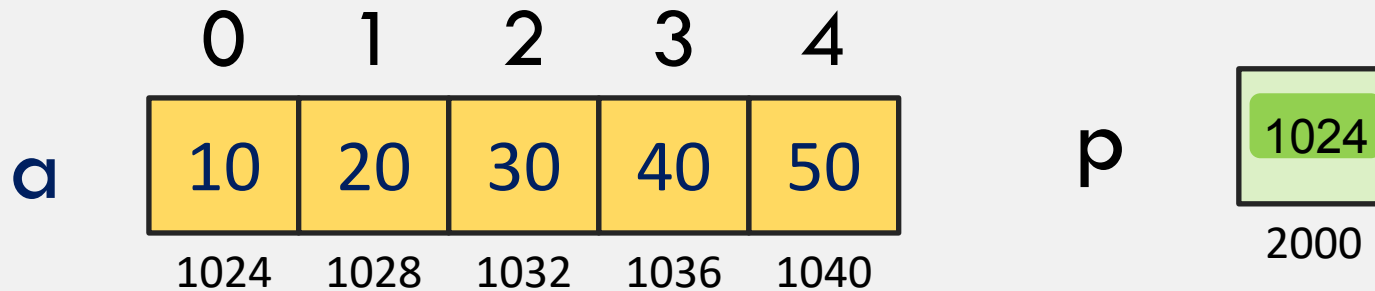
4.2 Pointer advanced

4.2.1 Pointer and array

```
int a[5]={10,20,30,40,50};
```

```
int *p;
```





a = 1024

a+1 = 1028

a[0] = 10

a[1] = 20

*a = 10

*(a+1) = 20

*a+1 = 11

&a[0] = 1024

a++

p = a

p++ = 1028

*p = 20

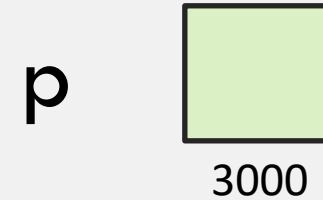
*(p++) = 30 (ต่อ

ของเก่า)

ห้ามขยับ array



	0	1	2	3	4
a	5.2	3.6	12	4.5	2.8
	2000	2004	2008	2012	2016



```
float a[5]={5.2,3.6,12,4.5,2.8};
```

```
float *p=a;
```

a	=	2000	*a	=	5.2
a+2	=	2008	*(a+2)	=	12
a[0]	=	5.2	*a+1	=	6.2
a[3]	=	12	*(a++)	=	12
			&a[3]	=	2008

p	=	a	=	&a
p++	=	3.6		
*p	=	5.2		
*(p++)	=	3.6	(ต่อของเก่า)	



ตัวอย่าง 9.3

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{    char array[10];
```

```
    char *ptr;
```

```
    array="hello"; /*wrong */
```

```
    ptr = "hello";
```

```
    cout << ptr;
```

```
}
```



array

**ตัวอย่าง 9.3**

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{    int *p ,*temp;
```

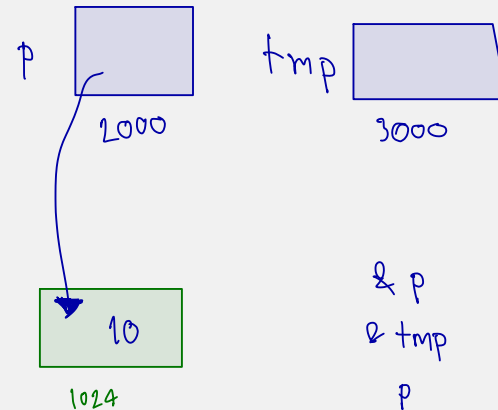
```
    p=new int;    สร้างกล่องใหม่ →
```

```
    temp=1024p; ย้ายมาอยู่ที่ 1024 มีค่า p ที่ 1024
```

```
    *p=10;
```

```
    cout << *temp; 10
```

```
}
```





4.3.2 Pointer to Structure

```
struct record
```

```
{    int id;
```

```
    float grade;
```

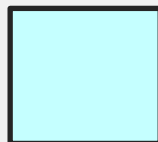
```
};
```

```
struct record std1, std2;
```

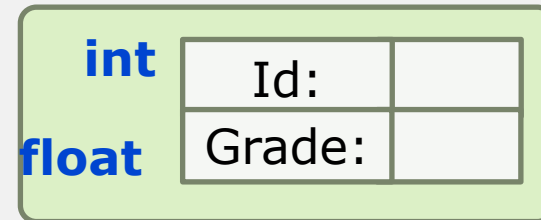
```
struct record *ptr;
```

& ptr 2000
* ptr 3018

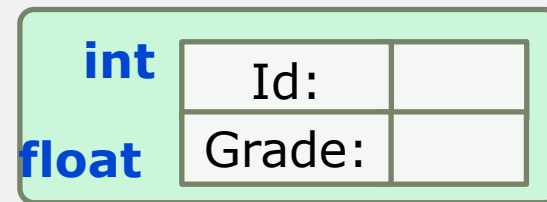
ptr



2000

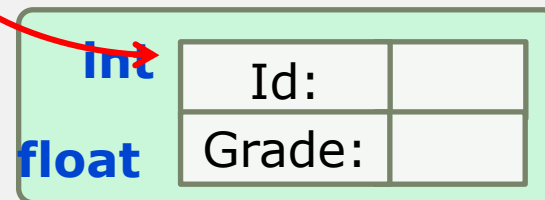


std1



3018

std2



3040



```
struct record std1, std2;
```

```
struct record *ptr;
```

```
std1.Id=100;
```

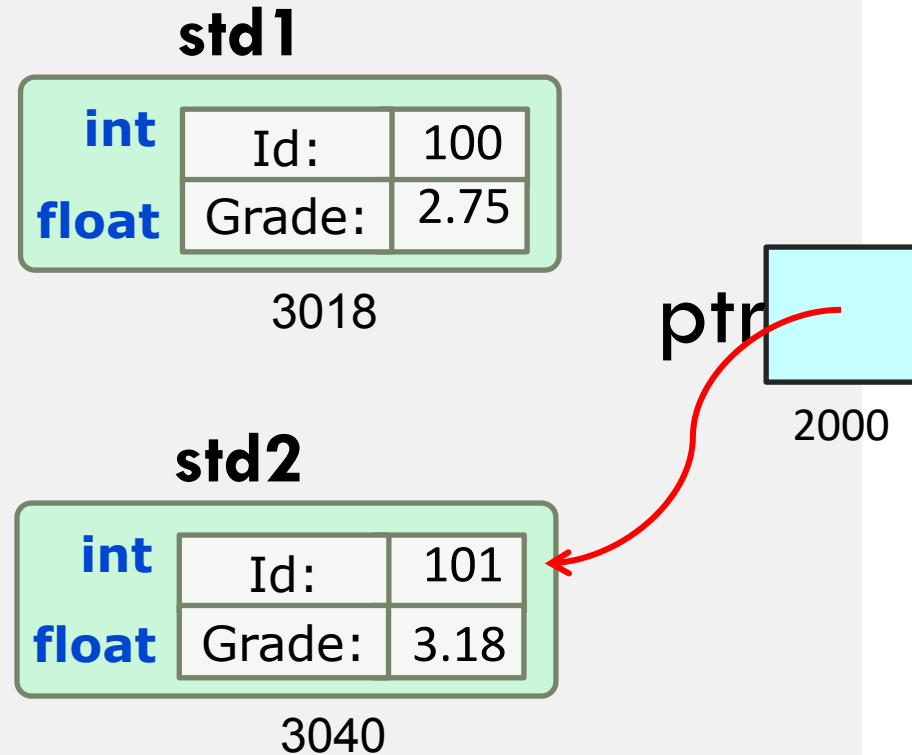
```
std1.Grade=2.75;
```

```
ptr=&std2;
```

```
(*ptr).Id=101; หรือ
```

```
ptr->Id=101;
```

```
std2->Grade=3.18;
```



ptr -> Id = 101

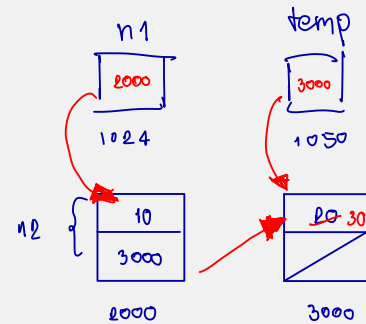
ptr -> Grade = 3.18



03603212 : Module1 – Introduction 35

```
#include <iostream>
using namespace std;
struct record
{
    int value;      4
    struct record *next;  8
};
int main()
```

```
{
    struct record *n1,*temp;
    n1=new struct record;
    temp=new struct record;
    n1->value=10;
    n1->next=temp;
    temp->value=20;
    temp->next=NULL;
    cout << n1->value << endl;  10
    cout << n1->next->value;    30
}
```



next = 20

2.1 A Brief Introduction to Recursion :

Mathematical function

1. $C = 2(F - 32) / 9$

2. $f(x) = x * f(x - 1)$
 $f(1) = 1$

3. $f(x) = 2f(x - 1) + x^2$
 $f(0) = 0$, x nonnegative integer



Circular logic?



Factorial $x! = x * (x-1) * (x-2) * \dots * 1$

$$5! = 5 * 4 * 3 * 2 * 1$$

$$5! = 5 * 4!$$

$$f(x) = x * f(x-1)$$

4. $f(x) = x * f(x-1)$

$f(1) = 1$, x nonnegative integer

Example 1

```
#include <stdio.h>

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

```
int main()
{   int ans;
    ans = fact(3);
    cout << ans;
}
```

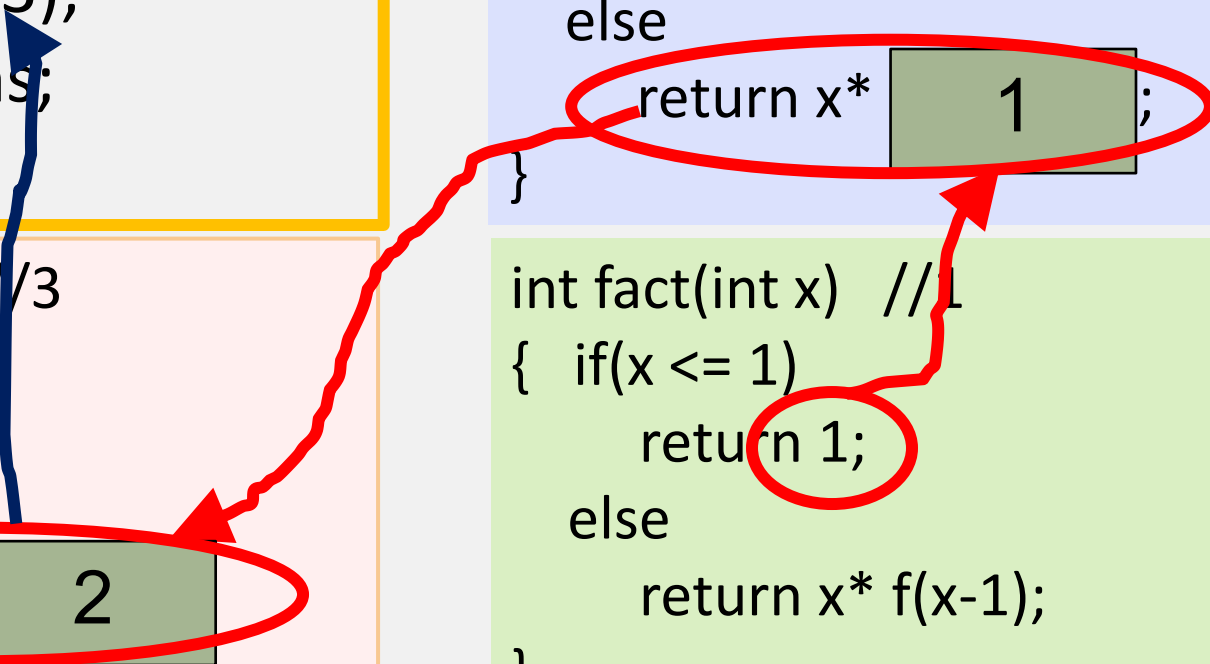
Example 2 Factorial

```
int main()
{
    int ans;
    ans = fact(3);
    cout << ans;
}
```

```
int fact(int x) //3
{
    if(x <= 0)
        return 1;
    else
        return x* 2;
}
```

```
int fact(int x) //2
{
    if(x <= 0)
        return 1;
    else
        return x* 1;
}
```

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





Example 3

```
int bad(int n)
{  if (n==0)
    return 0;
   else
    return bad(n/3 + 1+ n - 1);
}
```

Example 4

```
void printout(int n)
{  if( n >=10 )
    {
        printout(n/10);
    }
  cout << n%10;
}
```




03603212 : Module2–List, stack, Queue

```
void printout(int n)
{   if( n >=10 )
    {
        printout(n/10);
    }
    cout << n%10;
}
int main()
{   int n=1024;
    printout(n);
}
```