Computer 142

Alaa Itaimi

lecture 1 :

Computer:- A device that takes data as input- Process it and produces information as output

Difference between DATA and INFORMATION :-

Data: facts : البيانات التي تدخل وهي ليست مهمة بعض الشيء لانه لا يتم يتخذ عليها قرار

Information:- يتخذ قرار على

Computer

- Hardware : physical parts
- Software: logical Parts or set of instructio That tell hardwar what to do

As a summary : A hardware consists of :-

1) CPU : Central Processing unit (Brain of computer)
   - ALU: Arithmetic / logic units
   - CU: Control Unit
   - Register :- CPU وحدة تخزين داخل
2) storage < primary / secondary
3) Input / output

To explain it more :- ① CPU

a) ALU

Arithmetic : الحاسوب يستطيع القيام بعمليات جمع فقط

جمع = طرح = ضرب = قسمة

logic :- > or < or = : Basic logic

b) CU : controls everything

c) Register :- you put things that you want to reach it fast But it's not that big (it fits important things)

CPU = CU + ALU

The Machine cycle (4 processes) :-

1- fetch (get) next instruction
2- decode : ترجمتها : find what instruction is (changes instruction to commands)
3- Execute : run instructions
4- Store in RAM

② Storage :
- → Primary :- RAM
- → secondary : Disks, flash, Cd, dvd, ...

③ Input & output

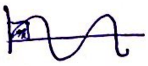
Input	Output
Keyboard	Printer
scanner	Speakers
Michrophone	Screen
	Plotter ← الخرائط (طابعة كبيرة)

Difference Between RAM & ROM:-

RAM	ROM
① Read & write	Read only
② Volatile متطاير = Temporary	Not-volatile غير متطاير

Lecture 2

- Data Representation , Data can be :-

- numbers → integers / → floats
- Charachters → ASCII   A → 65   a → 97
- Pictures  ..
- videos
- sound → 
- stings "ali"

Bin. To represent data, we change it to numbers and in the Decimal system. Those numbers we represent them as in the Binary system

1: on   0: off

Numbering systems

- binary :- (الثنائي) 2   (0-1)
- Decimal :- (العشري) 10   (0-9)
- octal :(الثماني) 8   (0-7)
- Hexadecimal (الستعشري) 16 (0-15) (0,-9, A, B, C, D, E, F)

How to change from decimal to binary :-

Exp:- $(23)_{10} \longrightarrow (10111)_2$

① 23 ÷ 2 = 11, 22, remainder 1

② 11 ÷ 2 = 5, 10, remainder 1

③ 5 ÷ 2 = 2, 4, remainder 1

④ 2 ÷ 2 = 1, 2, remainder 0

⑤ 1 ÷ 2 = 0, remainder 1

How to chang from binary to decimal

$(10111)_2 \rightarrow (23)_{10}$

$2^4 \; 2^3 \; 2^2 \; 2^1 \; 2^0$

$16 + 0 + 4 + 2 + 1$

$= 23$

How to change from decimal to octal

$(23)_{10} \rightarrow (27)_8$

① $23 \div 8 = 2$, 16, remainder 7

② $2 \div 8 = 0$, remainder 2

How to Change from octal to decimal

$(27)_8 \rightarrow (23)_{10}$

$8^1 \; 8^0$

$16 + 7$

• We change between any systems in the same Way

As an example :- $(23)_{10} \rightarrow (113)_4$

$23 \div 4 = 5$, 20, remainder 3 → $5 \div 4 = 1$, 4, remainder 1 → $1 \div 4 = 0$, remainder 1

$(113)_4 \rightarrow (23)_{10}$

$4^2 \; 4^1 \; 4^0$

$16 + 4 + 3$

How to change from Hexadecimal to decimal

$(29)_{10} \rightarrow (1D)_{16}$

$29 \div 16 = 1$, 16, remainder 13 (D)

$13 \div 16 = 0$, remainder 1

Note :-

10 → A

11 → B

12 → C

13 → D

14 → E

15 → F

$(1D)_{16} \rightarrow (29)_{10}$

$16^1 \quad 16^0$

$16 + 13 \rightarrow 29$

Examples:- Between 2 systems that we don't know

$(12)_3 \rightarrow (10)_5$

$3^1 \quad 3^0$

$3+2$

$(5)_{10}$

$5 \overline{)5} = 1$, remainder 0 $\rightarrow$ $5 \overline{)1} = 0$, remainder 1

• Examples

① $(AB37CDF245)_{16} \rightarrow (\quad)_8$

$16 \rightarrow 2^4$ $8 \rightarrow 2^3$

001 010 101 100 110 111 110 011 011 111 001 001 000 101

$(1\ 2\ 5\ 4\ 6\ 7\ 6\ 3\ 3\ 7\ 1\ 1\ 0\ 5)_8$

→ This works only when the system can be divided on 2 like 8, 16, 32 ...

Note:-

$2^3\ 2^2\ 2^1\ 2^0$

0000 → 0
0001 → 1
0010 → 2
0011 → 3
0100 → 4
0101 → 5
0110 → 6
0111 → 7
1000 → 8
1001 → 9
1010 → 10
1011
1100
1101
1110
1111 → 15

② $(37245)_8 \rightarrow (3EA5)_{16}$

$8 \rightarrow 2^3$ $16 \rightarrow 2^4$

$(011\,111\,010\,100\,101)_2$

$(2.5)_{10} \rightarrow (10.1)_2$

$0.5 \times 2 = 1.0$

$(2.25) \rightarrow (10.01)_2$

$0.25 \times 2 = 0.5$

$0.5 \times 2 = 1.0$

لو كان الجواب
مثلاً 1.6
بأخذ 0.6 فقط

lecture 3

ex:- $(52*)_8^{2^3} \rightarrow (222)_4^{2^2} \rightarrow (46)_9$

$\hookrightarrow (101010)_2 \rightarrow (42)_{10}$ ($2+8+32$; $\times 4^2, \times 4^1, \times 4^0$)

$42 \div 9 = 4$ remainder 6 ; $4 \div 9 = 0$ remainder 4

- To make sure that the answer is right we change the octal to decimal, then we change the ( )$_4$ to decimal & if the answer is 42 for all then My answer is right!

Examples :-

① $(13.125)_{10} \rightarrow (1101.001)_2 \rightarrow (13.125)_{10}$

$2^3\ 2^2\ 2^1\ 2^0 \cdot 2^{-1}\ 2^{-2}\ 2^{-3}$

(من نقطة الأعداد .)

$8+4+0+1 \cdot (0+0+\frac{1}{8})$

$13 \quad 0.125$

$0.125 \times 2 = 0.25$

$0.25 \times 2 = 0.5$

$0.5 \times 2 = 1.0$

To change from 2 to 8 :-

$(001\,101 \cdot 001)_2$

$\hookrightarrow (15 \cdot 1)_8$

from 2 to 16

$(1101 \cdot 0010)_2$

$(D \cdot 2)$

$16^{+0}\ 16^{+16}$

$= (13.125)_{10}$

• Binary operations

$$\begin{pmatrix} 101 \\ 10 + \\ \hline 111 \end{pmatrix}_2 \quad \begin{pmatrix} 5 \\ 2 + \\ \hline 7 \end{pmatrix}_{10}$$

• Rules:- • نجمع رقمين فقط

$$\frac{0}{0}+ \quad , \quad \frac{0}{1}+ \quad , \quad \frac{1}{0}+ \quad , \quad \frac{1}{1}+ \quad , \quad \frac{1}{1}+$$

Examples:-

① $\begin{array}{r} 1011 \\ 10000 + \\ \hline 11011 \end{array}$

② $\begin{pmatrix} 11011 \\ 11100 + \\ \hline 100111 \end{pmatrix}_2 \quad \begin{pmatrix} 11 \\ 28 + \\ \hline 39 \end{pmatrix}_{10}$

③ $\begin{array}{r} 1\ 1 \\ 111 \\ 111 + \\ \hline 1110 \end{array}$

④ $\begin{pmatrix} 11.01 \\ 101.10 + \\ \hline 1000.11 \end{pmatrix}_2 \quad \begin{pmatrix} 3.25 \\ 5.5 + \\ \hline 8.75 \end{pmatrix}_{10}$

⑤ $\begin{pmatrix} 11.11 \\ 10.10 \\ \hline 110.01 \\ 6 \quad 0.25 \end{pmatrix}_2 \begin{pmatrix} 3.75 \\ 2.5 \\ \hline 6.25 \end{pmatrix}_{10}$

Note :-

+ ← 0101 : اذا كان الرقم يبدأ بـ 0 هو موجب

- ← 1101 : اذا كان الرقم يبدأ بـ 1 هو سالب

Two's complement

Negative: ←101→ ~~5~~ It's -3

Positive: ←0101→ +5

- 1's complement : 101 → 010
  - we put 0 instead of 1 and we put 1 instead of 0
- 2's complement
  - we add 1 to the number

010 + 1 = 011 (+3)

↳ so 101 is -3

Ex: 0110 → +6

1001 +1 = 1010 → -6

0101 +1 = 0110 → +6

5 - 2

= 5 + (-2)

0101 , 1110

0101 + 1110 = 10011

- 2 is 0010 ↳ +2

to get -2 we use 2's complement :-

0010 → 1101 +1 = 1110

- لا نأخذ ١ على اليسار لأنه العملية كانت بين رقمين اكبرهما تتكون من اربع خانات وبالتالي الجواب يتكون من اربع خانات ونستثني الخانات من اليسار

Ex: (-7)+1

(1001) (0001)

-7 من

1001
0001 +

1010 ← -6

+7 من 0111
1000
1 +

1001

• -7 +
1

-6

Ex: ~~6 3~~

-5 - 1

1011 1111

1011
1111 +
---
1 1010 ← -6

• +5 من 0101
1010 +
1

-5 ← 1011

• +1 من 0001
1110
1 +

-1 ← 1111

Ex: -5 - 4

1011 1100

1011
1100 +
---
1 0111 ← +7

• +4 من 0100
1011 +

1011 +
1

1100 ← -4

-5 +
-4

-9

• الجواب خاطئ لان الارقام التي دخلت في العملية تحتاج الى اربع خانات لتمثيلها
ولكن الجواب (-9) وهو الصحيح يحتاج الى خمسة خانات لتمثيله

• لذلك الحاسوب يستعمل عدد خانات كثير لكي يتأكد من الجواب فيكون عدد الخانات 64 و 128 او عدد اكبر من الخانات

• عندما يكون الجواب غريب او خاطئ نزيد عدد الخانات للتأكد

[Le]cture 4:-

[Exa]mple:- Using two's complement with 8-bits Solve :-

① $(75)_{16} - (34)_{7} = (\cancel{1}01011100)_2 = (1130)_4$

$(75)_{16} \rightarrow (01110101)_2$

$(34)_7 \rightarrow (25)_{10} \rightarrow (00011001)_2$

```
 11100110
        1 +
 ________
(11100111)_2
```

$(01110101)_2 + (11100111)_2 =$

```
  01110101
  11100111 +
 _________
(\cancel{1}01011100)_2
```

② $(20)_{10} - (15)_{10} =$

$(00010100)_2 - (00001111)_2 =$

```
 11110000
        1 +
 ________
(11110001)_2
```

$00010100_2 \cdot (11110001)_2$

```
  00010100
  11110001 +
 _________
 \cancel{1}00000101
```

Data Presentation :-

1. integer : 16 Bit ⇒ $2^{16} = 65536$ (15 Bits; This bit is for signs)

   -32768 ... 0 ... 32767

   -2^{15} → $+2^{15}-1$

2. float :
3. char :

- 2' complement is used with integers

If I had one bit I can put either 1 or zero so $2^1 = 2$

If I had 2 bits I can put 4 diff things $2^2 = 4$

$(37)_{10} = (100101)_2$

high order → 0000 0000 0010 0101 ← low order

0 0 2 5

- How is it saved in the memory?

memory 8 bit
25
00

→ small Endian

- There is 2 ways : either the Big Endian or the small Endian

$(-17)_{10}$

+17 $(010001)_2$

$(0000000000010001)_2$

1111111111101110

1 +

1111 1111 1110 1111

F F E F

Memory

EF
FF

③ Char

ASCII → Extended ASCII → Unicode (Universal Code)

a-z , A-Z , 0-9 , ? , #, ----

- ASCII (128)

↳ American standard Code for information interchange

A - 65 d - 100

a - 97

d → memory → using even or odd parity

↳ $(100)_{10}$ → $(\_1100100)_2$ | Mem 64 |

parity bit (error detection)

↙ odd ↘ even

if using odd: 0 1100100
There is Three 1 (it's an odd no)
So We put 0 in The parity bit

if using even 1 1100100
There is Three 1 (it's an odd no)
So We put 1 and it becomes 4 which is a parity bit

As an example ahmad :-
a →
h →
m →
a →
d →

a - 97
b - 98
c - 99
d - 100
e - 101
f - 102
g - 103
h - 104
i - 105

② float :-

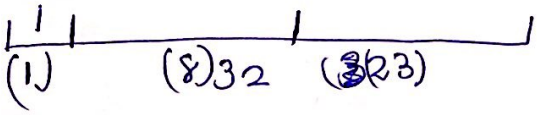
5247.32
The Scientific No.
is 5.24732×10^3

$(-4.25)_{10}$

↳ $(100.01)_2$ ↙

- Scientific notation

$= 1.0001 \times 2^2$



Lecture 5

Floating Point representation

$(-4.25)_{10}$

① 100.01

② 1.0001 × 2^{2}

الأخزن

$2^8 = 256$

C	0	8	8	0	0	0	0

1 | 10000001 | 00010000000000000000000

sign — 8 bit exponent — 23 bit mantissa

الجزء الكسري من الرقم

127 — 128 → zero, 126

اعتبر 127 الصفر

$\Rightarrow 127 + 2 = 129$

Memory

0 0
0 0
8 8
C 0

$(0.2) \times 2 = 0.4$
$(0.4) \times 2 = 0.8$
$(0.8) \times 2 = 1.6$
$(0.6) \times 2 = 1.2$
$0.2 \times 2 = 0.4$

① 0.00110

② 1.10×2^{-3}

$= 1.① \times 2^{-3}$

$127 + (-3) = 124$

$2^2 + 2^0 = 3$ بنقص منه 127 أو نحل

3	E	4	0	0	0	0	0

0 | 01111100 | 10000000000000000000000

sign

Mem

0 0
0 0
4 0
3 E

• Algorithms : طرق الحل

- sequence
- Conditional
- Repetition

• in Algorithms There is an input, an output & processing

find The Algorithm to sum any Two numbers

input:
- Ask user to enter first number
- Read number and save as num1
- Ask user to enter second number
- Read number and save as num2

processing:
- Add num1 to num2 and save result as sum

output:
- Print sum to screen

• دائماً البدأ بفعل أمر
• الجملة يجب ان تكون سطر واحد (١٠ كلمات كحد أعلى)
• عكس استعمال أي نوع من الكلمات بشرط ان يكون المعنى كامل وصحيح
• الترتيب مهم أحياناً

In The Top-down design :-

1- get number → 1.1 get first number → 1.1.1 Ask user / 1.1.2 ~~Ask~~ Read --
→ 1.2- get second number

2- first sum

3- Print sum

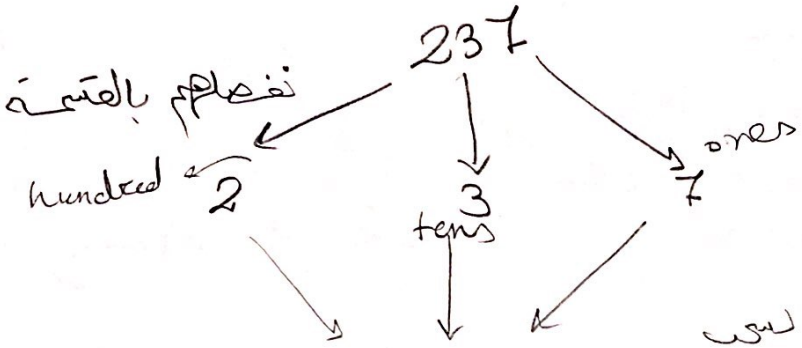
lecture 6

Write an algorithm (pseudo-code) :-

- to reverse any given three digit number

237 → 732

102 → 201



لنجيب المئات . نقسم على ١٠٠

$$\begin{array}{r|l} 100 & 237 \\ & 200 \\ \hline & 37 \end{array} \quad \boxed{2}$$

لنجيب العشرات . نقسم على ١٠

$$\begin{array}{r|l} 10 & 237 \\ & 230 \\ \hline & 7 \end{array} \quad \boxed{23}$$

لنجيب الآحاد . نقسم على ١٠

$$\begin{array}{r|l} 10 & 237 \\ & 230 \\ \hline & 7 \end{array} \quad \boxed{7}$$

732

5%2 = 1 هو الباقي

2%8 = 2

Bul

512 . 2

. لنجيب العشرات هناك أكثر من طريقة :

١- نأخذ آخر خانتين 37 ونقسم على ١٠ ونأخذ الكوتا

$$\begin{array}{r|l} 10 & 37 \\ & 3 \end{array}$$

٢- أو نأخذ أول خانتين ونقسم على ١٠ ونأخذ الباقي

- The Algorithm:-

1- Ask user to Enter any three digit number] inpi
2- Read number and save as num
3- Divide num by a hundred and save result as hunds
4- Divide num by ten and save remainder as ones
5- Divide num by a hunds and save remainder as temp
6- Divid temp by ten and save result as tens

7- Multiply ones by a hundred and. Save Result rev

8- Multiply tens by ten and add result to rev

rev += ten * 10
rev = rev + (tens * 10)

9- Add hunds to rev

10- Print rev to screen

We Try :- 352

num = 352

hunds = 352/100 = 3

ones = 352 % 10 = 2

temp = 352 % 100 = 52

tens = 52 % 10 = 5

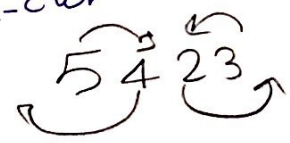
rev = 2 * 100 = 200

rev = 200 + (5 * 10)
= 250

rev = 250 + 3
= 253

253

Example :- In-out



Ex: 5123 % 10 = 3
5123 / 10 = 512

512 % 10 = 2
512 / 10 = 51

51 % 10 = 1
51 / 10 = 5

5 % 10 = 5
5 / 10 = 0

x = num % 10 ;
num = num / 10 ;

x = num % 10 ;
num = num / 10 ;

x = num % 10 ;
num = num / 10 ;

We use the loop

Conditional (selection)

num → neg
num → pos

$>$
$<$
$==$

X = 5   put 5 in X
X == 5   does 5 equal X

• If statement

If num is less than zero
تعبير الظرف ⟷ Print "num is negative" to screen

Else
Print "num is positive" to screen

END IF

Conditional (Selection)

lecture 7

- Write an algorithm to decide whether a given number is odd or even

Ask user to enter any number
Read number and save as num
Divide num by two and save remainer as rem
If rem equals zero
    Print "num is even" to screen
Else
    Print "num is odd" to screen
End If

- Write an algorithm to change marks to letter grades such that (A = 90-100, B = 80-90, C = 70-79, D = 60-69, F = 0-59)

Ask user to enter mark
Read mark and save as mk
IF mk is greater than or equal to ninety
    Print "grade is A" to screen
    Print "good job" to screen
Else If mk is greater than or equal to eighty
    Print "grade is B" to screen
Else If mk is greater than ~~or~~ sixty nine
    Print "grade is C" to screen
Else If mk is greater than or equal sixty ~~nine~~
    Print "grade is D" to screen
Else ~~If mk is greater than~~ Print "grade is F" to screen
Print "see you ~~next time~~ semester" to screen

Else

Print "grade is F" to screen

Print "see you next semester" to screen

End If

To save the grades and use them in another thing we Type after every Else If

set grade to 'A'
equal

X = A and X = 'A' is different

This means That A is a variable. The computer will search for a of A

This means That (A is a constant)

- If ____

Else If ____

End If

اذا تحقق الشرط اكمل كذا
اذا لم يتحقق اكمل هيك

If ____

Else

If ____

End If

End If

- اذا لم يتحقق الشرط يدخل على If أخرى ويرى اذا تحقق الشرط أم لم يتحقق

Note:-

* Conditions can have : And, Or, Not in if

Example:

X	y	X and y	X or y	Not X
T	F	F	T	F
F	T	F	T	T
T	T	T	T	F
F	F	F	F	T

• If mk is greater than eighty nine and mk is less than or equal to a hundred

Print "greater is A" to screen

End If

Repition:- (loops)

There is Three Types of loops:-

- While
- do/while
- for

• Write an algorithm (pseudo-code) to find & print the avarge grade for a class of ten students

- Set sum equal to zero
- Set count equal to zero

لوضرب بحب one

• Two things change
- sum
- count

- Write an algorithm to calculate & print the avereage grade for a clan of ten students

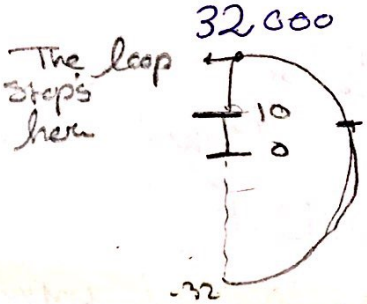
Set sum equal to zero → initial value
set Count equal to zero
while Count is ~~not~~ less than ten → final value (condit
Ask user to enter grade → Read grade and save as grade
Add grade to sum
Increment Count by one → change (التغيير الى)
END while
Divide sum by ten and save result as avg
Print avg to screen

- How does the computer Work :-

Count = 0 ? <10 yes
gr = 50
sum = 0 + 50 = 50
Count = 1 ? <10 → yes
gr = 70
sum = 70 + 50 = 120
count = 2
⋮
count = 9
gr = 6
sum = 770 + 6 = 776
Count = 10
avg = 776 / 10
= 77.6

Note

- If you put decrement by -1 it makes a loop of 32,000



- Write an algorithm to find the avg grade for a class with an unspecified number of students (0–100)
  ↑ Range

```
set sum equal to zero
set count equal to zero
Ask user to enter grade or -1 to stop
Read grade and save as gr
While gr is not equal to -1
    Add grade to sum
    Increment count by one
    Ask user to enter grade or -1 to stop
    Read grade and save as gr
END While
If count is greater than zero
    set avg equal to sum divided by count
    Print avg to screen
Else
    Print "No grade entered" to screen
ENDIF
```

Priming the pump

- للعلامة التي نعرفها ونفحص مباشرة

~~Sentinel~~

- sentineel:
إشارة تدل على التوقف عن العد
عندما نعرف عدد اللفات
- when we don't know the times of loop we use a way called priming the pump

- هنا الـ Sentineel لها علاقة بالـ input لأنه الـ Range محدد

Write an algorithm to find the average grade for a class with an unspecified number of students

Set sum equal to zero
Set count equal to zero

Ask user to enter whether to continue or not (y/n) → yes → no

Read answer and save as ans → ال user بدخل ال initial value

While ans is not equal to 'n' → final value

Ask user to enter grade
Read grade and save as gr
Add gr to sum
Increment count by on → change
(The rest is the same)

End while

* هنا ال sentinel ليس له علاقة بال input لانه ال Range غير معروف

IN the Computer :-

```
#include <stdio.h>
int main( )
{
    int n1, n2;
    int sum ;
```

lecture 9

1 Ask user to enter first number
2 Read number and save as num1
3 Ask user to enter second number
4 Read number and save as num2
5 set sum equal to num1 plus num2
6 Print sum to screen

```
#include <stdio.h>
int main()
{
    int num1, num2;
    int sum;
    printf("Enter first number \n");
    scanf("%i", &num1);
    printf("Enter second number");
    scanf("%d", &num2);
    sum = num1 + num2;
    printf("sum = %d \n", sum);
    return 0;
}
```

#include <stdio.h> → standard input/output header file

int main() → always exists, you have to know them by heart

{ → integer

int num1, num2; → data Type; variables

int sum;

printf("Enter first number \n") → \n: this means go to the next line

scanf("%i", &num1);

printf("Enter second number");

scanf("%d", &num2); → stops the program; & address: عنوان واسم الخل الذي سوف توضع به القيمة

sum = num1 + num2; → Assignment statement

printf("sum = %d \n", sum);

return 0; → success

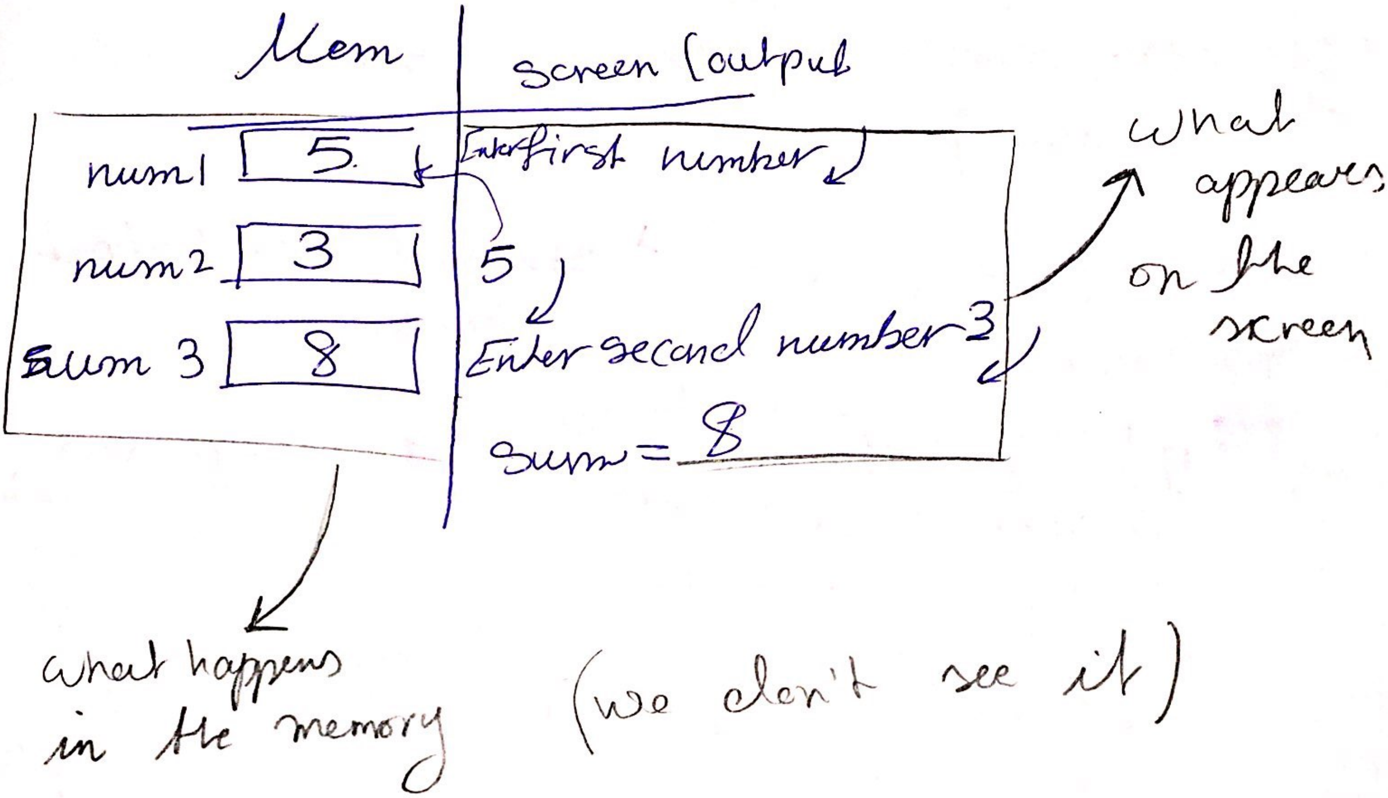
important: Search for Camle Notation

Notes :-
1- Since num1 and num2 is the same type we can put them in one line and we end it with (;). If they were different we split them with (;)

2- Variables name can include letters + numbers + underscore But it can't start with a num s.a 2nd → wrong / First_num → right

→ you can't name variables with key words (reserved words) such as main, int .. (words that exist in the basic sentences of the program

• In The Memory :-



• On the computer :-

compile (build) → Code Block
run (link) → Code Block

• If u made a mistake it either shows an error or a warnings

- error → you can't run the program So you should have 0 errors
- error → logical error, syntax ملاحظة! , run-time errors s.a: number/0 عدد --- صفر
- warnings → Try to make them 0 warnings. sometimes they don't affect your program But you should correct it Bcz sometimes it's dangerous to leave them

Examples of errors

① X=5 ;
y = X-4;
y=y-1;
z=3/y; ⟹ run-time error

② if the program is to sum and input — instead of
+ ⟹ logical error

③ avg=(g1 + g2)/2 ; (g1: 50, g2: 100) ↷ you should put ( )
avg → 100 ⟹

lecture 10

- Constants

```
#include <stdio.h>
#define PI 3.14
int main ( )
{ int rad;
  float area;
  Printf ("Enter radius\n");
  scanf ("%d", &rad);
  area = PI * rad * rad;
  Printf ("Area=%f", area);
  return 0;
}
```

→ constants

- usually we use All capitals for constant
- Constant ␣ its value (No =)
  (space)

- Types
- To read and write X :-

If x is	int	float	double	char
	scanf("%d", &x); Printf("%d", x);	scanf("%f", &x); printf("%f", x);	scanf("%lf", &x); printf("%f", x);	scanf(" %c", &x); (space) printf("%c", x);

Example

```
int age;
char gender;
printf ("Enter age and gender /n");
scanf ("%d %c", &age, &gender);
```

↳ we put a space so it doesn't take the charcter "space"

- when you enter you put a space if the 2 numbers are float

- binary operations

Arithmetic operations

+	3+2=5
−	3−2=1
*	3*2=6
/	3/2=1
%	3%2=1 , 6%10=6

Preceelence Rules

$X = 5 + 2*3 + 4/2 * 5$

$= 5 + 6 + 2*5$

$= 11 + 10 = 21$

* / % } stronger
+ − : weaker

$X = [(5+2) * (3+4)] / (2*5)$

$= (7 * 7) / 10 = 4.9$

Type casting

```
int   X = 3, y = 2; a;
float Z = 2.4;

X = 3/2;   → 1
Z = 3/2;   → 1.0
X = 3/2;   → 1
Z = 3/2.0; → 1.5
```

لكي تحصل على الجواب الكسر
في ① اما يكون البسط او المقام
او الاثنين مكتوبا به كـ float
مثل 2.0
② في اما يكون العكاد الذي
سنضع فيه الجواب float
مثل Z

a = x/y; ⟶ 1

b = x/y; ⟶ 1.0

a = x/y.0; ⟶

↳ to make it float for once we use type casting

⇒ a = (float) x/y; ⟶ 1

z = (float) x/y; ⟶ 1.5

z = x / (float) y; ⟶ 1.5

z = (float) x / (float) y; ⟶ 1.5

z = (float) (x/y); ⟶ 1.0

int → z → float

Note : • we can use z = x; without type casting

But • x = z; ∴ you have to explain that you want z to become an integer

↓ int

• x = z % y;

↳ int

Example: $V = \frac{4}{3}\pi r^2 h$

= 4 /3.0* PI* r * r *h

↳ you have to put a zero

..output formatting

52␣3␣12345

2305 232

2␣345␣67

lecture 11

- output formating

x = 524 , y = 3 , z = 12345
= 2 , = 1245 , = 6
= 12345 , = 1 , = 713

Print f ("%d ␣ %d ␣ %d, x, y, z);

It will print

524␣3␣12345
2␣1245␣6
12345␣1␣713

If we put:

Print f ("%8d ␣ %7d%9d", x, y, z).

_ _ _ _ _ 5 2 4 | _ | _ _ _ _ _ _ 3 | _ _ _ _ _ 1 2 3 4 5

2   1245   6

12345   1   713

for integers and characteres

- اذا عدد الخانات 8 يكفي كله ... عدد الخانات لوحده على الحد

مثال: %3d %7d

|1 2 3 4|

عدد الـ 7 يبدأ من هنا

لو كان في space مثال:

%3d␣%7d

|1 2 3 4|␣␣

نضيف الـ space ونبدأ

لو كان عدد الخانات سالب نلزقه على الشمال

Print f ("%5d% -2d%-7d", x,y,z)

|_ _ 5 2 4 | 3 _ | 1 2 3 4 5 _ _

2 | 1 2 4 5 | 6 _ _ _ _

_ _ _ _ _

for a float   By default خانات أخذ

X = 73.624927   y = 5.261 , Z = 324.52

Printf ("%10.2f%7.1f %-8.4f", x, y, z);

width, عدد الخانات (تقريب), %.__f → width, number of digits

|______73.62|____5.3| 324.5200

• الرقم لحيث انه يكتب صحيحة ، اذا لم تكفي الخانات تكتب الرقم خارج الخانات

• Comments To write them there is 2 ways

① /* ___ */   /*——*/

```
#___
int___
{
```

or ② `sum = x+y;   //add x to y`

• files

```
#include <stdio.h>
int main()
{ int n1, n2, sum;
  FILE * in;
  in = fopen ("data.txt", "r");
  fscanf (in, "%d%d", &n1, &n2);
  sum = n1+n2;
  Printf ("sum=%d", sum);
  fclose (in);
  return 0;
}
```

• يعني هذا البرنامج يعتمد على Data موجودة في فايل اسمه data.txt

• يعني سجم لِؤشر لـ file (Pointer)

mode of the opened file — يعني هذا الملف سافتحه للقراءة

أعطت عنوان لـ in

بسبب وجود fopen نعرف المكان

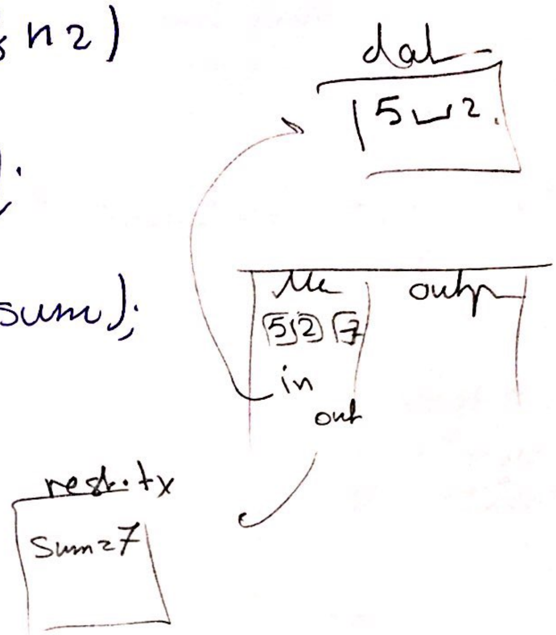
تلغي العلاقة بين in و data.txt

data.txt
5 2

Mem	output
5 (n1) 2 (n2) 7 (sum), in	sum = 7

```
#include <stdio.h>
int main ()
{ int n1, n2, sum;
File *in, *out;
in =fopen ("data.txt","r").
out= fopen ("rest.txt","w");
fscanf (in,"%d%d", &n1, &n2)
Sum = n1+n2.
Printf ("sum %d ",sum);
fprintf (out,"Sum=%d", sum);
fclose (out);
fclose (in);
return 0;
}
```

مهمة
يعني اذا كان
الشي غير
مكتوب
اكتبه واحفظه
بدونها يكون الملف فارغ
او معلومات غريب



lecture 12

- functions

Built starts here

```
#include <stdio.h>
int sum (int, int);
int main ( )
{
int x, y, s;
Printf ("Enter x and y \n");
scanf ("%d %d", &x, &y);
s = Sum (x, y);
Printf ("sum = %d", s);
return 0;
}
int sum (int a, int b)
{
int result;
result = a+b;
return result;
}
```

fun starts here

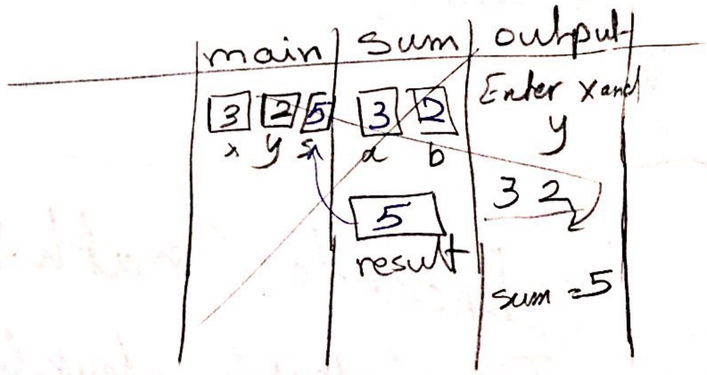
→ • function Prototype (اسمها، نوع الأشياء التي تأخذها، نوع الأشياء التي تخرجها)

مثلا: sum, int, int

s = Sum (x, y); → function Call لاستخدامها

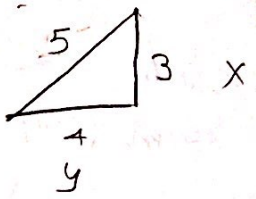
→ a small program to explain sum

int sum (...) { ... } → definition



Two Types

1) system defined functions
2) User defined functions



$$z = \sqrt{x^2 + y^2}$$

1) S.d.f :-

```
#include <stdio.h>
#include <math.h>
int main()
{
  int x, y;
  float z;
Printf ("Enter x and y\n").
Scanf ("%d%d, &x, &y);

z = sqrt( pow (x,2) + (y*y) );
Printf (" z = %.2f ", 2);
  return 0;
}
```

Include <math.h>

Pow (double, double) gives → double

sqrt (double); → double

floor (double); → int

ceil (double); → int

abs (int); → int

fabs (double); → double

sin(double) → double

cos(~) → ~

tan (~) → ~

sec (~) → ~

cos (~) → ~

Cot (~) → ~

2) U.d.f

.... sum (int, int) → int

sum ( ) → int

sum (int, int) → (Nothing) void

sum ( ) → (Nothing) void

lecture 13

Void: يأخذ ولا يرجع

Void

```
#include <stdio.h>
int sum(int, int);      -> if we put void (هذا الفنكشن لا يعطينا شيء)

int main()
{
  int x, y, ~~s~~;          we cancel s
  Printf ("Enter x and y\n");
  scanf ("%d %d", &x, &y);
  ~~s=~~ sum(x,y);
  ~~Printf ("sum=%d", s);~~

  return 0;
}
void ~~int~~ sum(int a, int b)
{
  int result;
  result = a+b;
  ~~return result;~~
  Printf ("sum=%d", result);
}
```

Question for void:-

2 numbers and it prints the two numbers (No output) (output is done by Sum Not main)

for the first (No void)

main	sum	output
2 4 6 x y s	2 4 6 a b	Enter x and y 2 4

main	sum	
2 4 x y	2 4 6 a b res	Enter x & y 2 4 sum=6

( )

```c
int sum( );
int main( )
{
    int s;
    s=sum( );
    Printf("Sum=%d", s);
    return 0;
}
```

sum( ) → يقوم ب Processing & input

main( ) → هو الذي يقوم ب output

( ) : لا يؤخذ ويرجع

مثل main( )

```c
int sum ( )
{
    int x,y, result;
    Printf ("Enter x and y\n");
    scanf ("%d%d", &x, &y);
    result = x+y;
    return result;
}
```

main	sum	output
6 (s)	2 (x), 4 (y), 0 (result)	Enter x and y sum=6

void f ( ) :-

```c
void sum( );
int main ( )
{
    sum( );
    return 0;
}
```

```c
void sum( )
{
    int x,y, result;
    Printf ("Enter x&y\n")
    scanf ("%d%d", &x, &y);
    result = x+y
    Printf ("sum=%d", result);
}
```

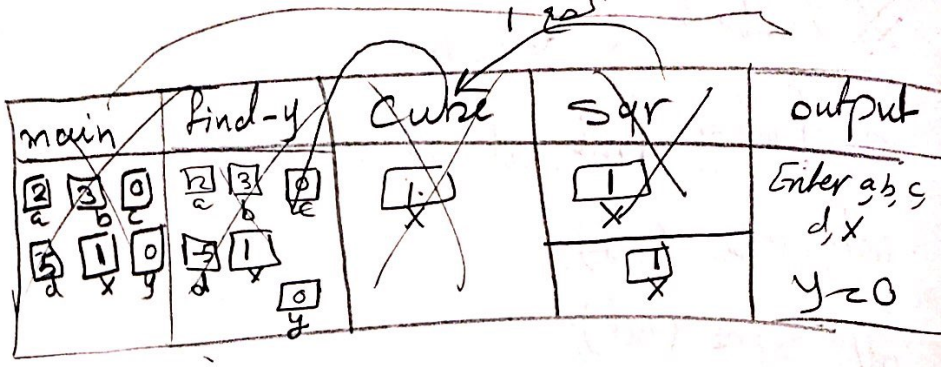
$pl: y = 3X^3 - 2X^2 + 0x + 5$ → a, b, c (0), d

y لطباعة ، x ، a,b,c,d برنامج يدخل

```
#Include <stdio.h>
int find_y (int,int,int,int,int);
int cube (int);
int sqre (int);
int  main ( )
{
 int a,b,c,d, x, y;
 Printf (" Enter a,b,c,d,x\n");
 scanf ("%d%d %d %d%d ", &a, &b, &c, &d, &x);
 y = finty ( a, b, c, d, x);
 Print f (" y=%d", y);
 return 0;
}
int find_y (int a, int b, int c, int d, int x)
{
 int y;
 y = a * cube(x) + b * sqr(x) + c * x + d;
 return y;
}
```

```
int cube (int x)
{
   return x*x*x * sqr(x) * x;
}

int sqr (int x)
{
   return x * x;
}
```

$2x^3 + 3x^2 - 5$



lecture 14

Selection

```
if (x>5)
  printf("good\n");
else
    printf("bad\n");
```

First :-
relational operators :-

<	less then
>	more (greater) than
<=	less or equal
>=	greater then or equal
==	للمقارنة ← (اما = هي معطى)
!=	not equal

Ex:.

```
if (x% 2 = =0)
    printf ("even\n");
else
  printf ("odd\n");
```

• Note:- [1] This is wrong
if (x=5) → should b ==
else

[2] In C languag
Zero → false
non-zero → true

But if you write it like this:-

```
if (X%2)
    printf ("odd\n")
else  "  " ("even\n")
```

• الحين هنا اذا قسم الرقم على 2 ← وكان الجواب صفر فانه تعتبرها zero / false وتذهب إلى else
اما اذا كان الجواب أي شي غير الصفر فانه تعتبرها True وتطبع odd

[3] If you write:-
X=2 ; X=0 , x=5

```
if (X=2)
   prinf("good\n");
else
   printf("bad\n");
```

It will print good for three cases

If you put:-
if (X=0)
It prints bad because zero = false In C

second:- logical operator :-

- &&	and
- ||	or
- !	Not

Ex1- if ((age > 20) && (gender == 'f') || (age > 85))

* it's either all of it is positive or all of it is false

X	y	x && y	x || y	!x
T	F	F	T	F
F	T	F	T	T
T	T	T	T	F
F	F	F	F	T

Ex age = 15, gender = 'f', avg = 70

if (F && T || !(F))

= (F && T || T) it starts with and

= (F || T)

= T

→ . !( ) is the most important
→ . && is more important than ||
→ . parenthesis ( ) is the most important

if (F && { T || !(F) })

(F && { T || T })

(F && T)

F

If it gi
if (x)
printf("good");
else
= ("bad");

Nawlf x=0
Then false
(It prints bad
If x is anythi
else It prints
good

200 (-

```
if (num >=0 );
   Printf ("%d is positive", num);

else
   Printf ("%d is negative", num);
```

Ex:
```
if (mark >=90); && (mark <=100)
{
      printf ("grade is A \n");
      printf ("good job \n");
}

else if (mark >=80)
      printf ("grade is B \n")

else if (mark >=70)
      Print ----- is C
else if (mark >=60)
      ---------- is F
```

*This means all the conditions should be True to take the result

• It's a one If so It goes to each condition and if one is True it stops.

Short circuits

```
x=5;   y=10;                   → if x=15
                 =20
if((x>6) && (y<20))
    F T              T
good
else
    bad
printf("%d \n", y);
```

Now if It was y=0 It is false so T && F = F It prints bad and 0

It will print good and 20

* a program to know wheather a letter is a vowel or not:-

```
Char letter;
Printf ("Enter letter \n");
Scanf (" %c ", &letter);
If (letter == 'a' || 'o' || 'i' || 'u' || 'e')
    Printf ("%c is a vowel, letter);
else
    Printf ("%c is not a vowel, letter);
```

Those are not zero so it's All T

This is wrong

you write : if (letter == 'a' || letter == 'o' || letter == 'i' || letter == 'i' || letter 'e' — —

— — — —

Lecture 15:-

```
if (num == 1)
    printf ("one\n");
else if (num == 2)
    printf ("two\n");
else
    printf ("No such Number\n");
```

- Questions:
  *Gives you switch and wants if for the same case or opposite

---

Using switch

- Notes
  - It works only with == (Doesn't work with < >)
  - Comes after or before ~ ~ ~ things that we know what (it doesn't work with float & string)
    ↳ s.a : integers & Ascci code (a, b, c --)

```
switch (num)
{
Case 1: printf ("one\n");
        break;
Case 2: printf ("two\n");
        break;
default: printf ("No such number\n");
}
```

→ break is important to stop the program

↳ after default need bcz you can put break but no the program ended any way

Exp: Vowels using switches:-

```
switch (letter)
{
Case 'a': Case 'i': Case 'o': Case 'e': Case 'u':
    printf ("%c is a vowel", letter);
    break;
default: printf ("%c is not a vowel", letter);
}
```

it doesn't to be to have here

- x y

```
if (x>y)
  printf ("%d is larger", x);
else
  printf ("%d is larges", y);
```

- x y z a b
- we assume that the first number entered is the min

x	y	z	a	b
5	2	4	3	2

↳ assuming x = min;

```
if (y < min)
  min = y;
if (z < min)
  min = z;
```

- you can't use switch in this case
- you need a loop

Ex : A program to print the number that is between

→ x y z — it has to print 2

x	y	z
2	1	7

```
if ((x>y) && (x<z)) || ((x>z) && (x<y))
    printf ("%d", x);
else if ((y>x) && (y<z)) || ( ' ' ' ' ' ' )
    printf ("%d", y);
else print ("%d", z);
```

Ex: Enter a formula & get the answer

input → 5+2

output 5+2=7

```
printf("Enter formula\n");
scanf("%d ␣ %c %d", &n1, &op, &n2);

switch(op)
{
Case (+) : result = n1+n2;
           break;

Case (-) : ______
           |
           |
}
printf("%d %c %d =%d", n1, op, n2, result);
```

on the screen
Enter formula
5+ 3
5+3=8

Nested if:-

```
x=5 , y=3;
if (x>y)
  if (x==5)
     printf("good\n");
  }
else
  printf("bad\n");
printf("bye\n");
  {
```

adding { }

else for the first if

• on the screen

			after adding
good	x=5, y=3 good	x=5 , y=3 good bye	x=5, y=3 good bye
—	x=5, y=8 bad bye	x=5, y=8 bye	x=5, y=8 bad bye
	bad bye	x=3 , y=5 bye	x=3, y=5 bad bye
bad bye	—	x=7, y=5 bad bye	x=7, y=5 bye

Lecture 16

Boolean function:

لا ترجع عدد function، لكنها ترجع جواب نعم أو لا (True or false)

```c
int isEven(int n)   // <- it's like asking is this even?
{
if (n%2==0)
    return 1;
else
    return 0;
}
```

```c
main()
{
 int x=5;
 if (isEven(x))
    printf("Even");
```

Another way

```c
int isEven(int n)
{
  if (n%2)
     return 0;
  else
     return 1;
}
```

The shortest way:-

```c
int isEven(int n)
{
   return !(n%2);
}
```

* loops :-

- while
- for
- do/while

• 3 Basic Components of loops:-

Ex:
```c
int x=2          // 1- initial value
while (x<=5)     // 2- final value
{
  printf("hi\n");
  x++;           // 3- change
}
```

2 hi
3 hi
4 hi
5 hi
6 ✗

Ex1 X=5;
X++;
printf("%d", X); it prints 6

Ex2 X=5;
++X;
printf("%d", X); it prints 6

Ex3 X=5;
y = ++X → Add 1 to X and Then do everything
printf("%d", X); it prints 6 But y = 6

Ex4 X=5;
y = X++; → Do everything Then add 1
printf("%d", X); 6 But y = 5

Ex5 X=5;
printf("%d", X++); it prints 5 Bec he does The print then he adds one.

Ex6 X=5;
Printf("%d", ++X); it adds one Then it Prints
So it prints 6

- you can use the ++, -- way for the operations : S.a:-

$X = X + 7; \equiv X += 7;$

$X = X * 2; \equiv X *= 2;$

$X++; \equiv X += 1;$

Ex:- factorial :-

: n!

5 ⟶ 5! ⟹ we need a loop !

5x4x3x2x1
or 1x2x3x4x5

i = n (initial value)

- In case your going down

```
while (i >= 1)
{
    result = result * i; // result *= i;
    i--;
}
```

→ result = 1
i = 5
→ result = 1*5 = 5
i = 4
→ result = 5*4 = 20
i = 3
→ result = 20*3 = 60
i = 2
→ result = 60*2 = 120
i = 1
→ result = 120*1 = 120

• in case your go[illegible] up

```
i=1
while (i <=5)
{
  result=result * i;   //result *= i;

  i++;
}
```

~~~~~~~~~~

Ex: x^y $2^3 = 2*2*2$

```
result=1;
i=1;
while (i<=y)
{
  result= result * x;

  i++;
}
```

$3^4 \Rightarrow x=3, y=4$

i=1
result = 1 * 3 = 3
i = 2
result = 3 * 3 = 9
i = 3
" = 9 * 3 = 27
i = 4
" = 27 * 3 = 81

• if i wanna make it as a function

```
int myPow (int x, int y)
{ int result = 1;
  int i = 1;
  while (i <= y)
```
~~~~~~~~~~

```
{
    result = result * x;
    i++;
}
return result;
}
```

int z = pow(7,5);
(5,7);

```
sum = 0;
i = 0;
while (i < 10)
{
    printf ("Enter grade \n");
    scanf ("%d", &grade);
    sum += grade;
    i++;
}
    avg = (float) sum / 10;

printf ("Enter grade or -1 to stop \n");
scanf ("%d", &grade);
while (grade != -1)
{   sum += grade;
}
```

Lecture 17

for loops

Types of loops
while
for
do/while

```
x=1;
while (x<=5)
{
    printf ("good\n");
    x++;
}
```

The same

```
for (x=1; x<=5; x++)
    printf ("good\n");
```

- If more than one variable controls the loop

```
x=1, y=10;
while ((x<=5) && (y>=5))
{
    printf ("hi\n");
    x++;
    y--;
}
```

in the same way:-

```
for ( x=1, y=10; ((x<=5) && (y>=5)) ; x++, y-- )
        printf ("hi\n");
```

---

x^y

```
result = 1;
i = 1;
while (i<=y)
{
   result *= x;
   i++;
}
Printf ("result=%d", result);
```

in the same way:-

```
for ( result = 1, i=1; i<=y; i++)
      result *= x;
Printf ("result = %d", result);
```

Ex:
```
for (i=1; i<=1000; i++ );
        Printf ("good\n");
```
this means don't do anything

• هنا بطبع good مرة واحدة

• تستعمل لعمل pause للبرنامج في حالة أردنا ان نعمل مدة زمنية بين كل output

do/while

برنامج لحسب عدد الخانات .

```
X = 12547;
sum = 0;
Count = 0;
While (X > 0)
{ digit = X % 10;
  X = X/10
  Count ++;
  sum = sum + digit
}
Printf (" Count = %d ", Count);
  "    (" sum = %d ", sum);
```

• How it works

X = 1254x
Count = 0

X = 1254
Count = 1

X = 125
Count = 2

X = 12
Count = 3

X = 1
Count 4

X = 0
Count = 5

If you want to add digits

Using do while:-

```
X = 12547;
Count = 0;
do
 { X = X/10;
   Count ++;
 } while (X > 0);
```

• نضعها لان الشرط جاء في أخر الحلقة

• do/while is the only one That you have to enter the loop for at least once

• Ex

• برنامج لطبع فردية العدد 100

```
i = 1;
While (i <= 50)      or (i < 100
```

```
{
 if (n%i == 0)   (100 written above n)
     printf ("%d\n", i);
   i++;
}
```

• برنامج بعطينا الأرقام الأولية

```
i = 2;
while (i<n)
{
    (n%i == 0)
       printf ("%d is not prime\n", n);
   else
     { printf ("  ——— is prime", n);
       i=n;}
}    i++;
```

• we use a Technique called: flag Technique

↳ you don't print till you make sure

```
Prime = 1;      (True)

i = 2 ;
while (i<n)
{ if (n%i == 0)
    prime = 0;
  i++;
}
```

lecture 18

break & continue :

```
X=3;
while (X<7)
{
  printf ("%d \n", X);
  if (X==5)
  break; continue
  X++;
}
Printf ("bye\n");
```

- لا تكمل اللفة بل ارجع افحص الشرط (it means → continue)

continue → it prints 3 4 5 5 5 ... infinite loop

break → it means اذا تحقق الشرط اوقف ال loop

= it prints 3 4 5 bye

Example :-

```
X=3;
while (X++<10)
{
  printf ("%d\n", ++X);
  if (X==7)
  break; continue;
  printf ("%d\n", X);
}
printf ("%d\n", ++X);
```

break - it prints:
5
5
7
8

* continue - it prints:
5
5
7
9
9
11
11
13

```
int i, prime, n;
prime = 1;        → for (n=1; n<=10000; n++)
for (i=2; i<n; i++)
    if (n%i ==0)
    {
```

```
        prime = 0;
        break;
      }
  if (prime)
    printf ("%d is prime", n);
  else printf ("%d is not prime", n);
```

Nested loop :- (added by)

. They don't have to be the same loop

Prime function :-

```
#include <stdio.h>
int isPrime (int);
int main ( )
{
  int i;
  for (i = 1; i <= 10000; i++)
    if (isPrimie(i))
      printf ("%d\n", i);
  return 0;
}
int isPrime (int n)
{
  int i = 2;
  while (i < n)
  {
    if (n % i == 0)
      return 0;
    i++;
  }
  return 1;
}
```

```
for (i=2; i<n; i++)
  if (n % i == 0)
    return 0;
return 1;
```

Example: Give me the perfect numbers (number = مجموع قواسمه
ماعدا ل [illegible] نفسه

مثال 6 = 1+2+3
22 = 1+2+4+7+14

Example:-

n = 4;

```
*
* *
* * *
* * * *
```

To print those :-

```
for (i=1; i<=n; i++)
    printf ("*");
```

To print

```
* * *   ← Rows
* * *   ←
* * *   ←
↑ ↑ ↑
colums
```

```
for (i=1; i<=n; i++) i
{
   for (j=1; j<=n; j++)
      printf ("*");
   printf ("\n");
}
```

i=1
* * *
i=2
* * *
i=3
* * *

To print

```
*
* *
* * *
```

(Added by)

To print

```
* * *
* *
*
```

To print

```
  - - *
  - * *
  * * *
```

```
for (i=1; i<=n, i++)
{ for (j                )
    printf (" ");
  for (k                )
    printf ("* ");
  printf ("\n");
}
```

Lecture 19

```
   *
  * *
 * * *
* * * *
```

```
for (i=1; i<=n; i++)
{
  for (j=n; j>1; j--)
    printf(" ");
  for (k=1; k=2*(i-1); k++)
    printf("*"); printf("%d",i)
  printf("\n");
}
```

```
    1
   2 2 2
  3 3 3 3 3
 4 4 4 4 4 4
```

you print K

```
    1
   1 2
  1 2 3
 1 2 3 4
```

```
    1
   2 2
  3 3 3
 4 4 4 4
```

→ you print i

printf("%d",i) instead of printf("*")

files and loops

```
int status;
status = scanf("%d", &x);
```

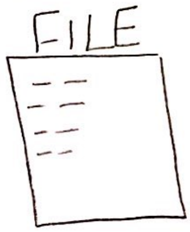
- you can use scanf to stop (when something is odd and not known)

```
-1
status = fscanf(in,"%d", &x);
```

• برنامج يستقبل علامات من File و يتوقف عندما نطلب منه ذلك

```c
float avg;
int g1, g2, status;

FILE *in;
in = fopen("data.txt", "r");
status = fscanf(in, "%d%d", &g1, &g2);
while (status != EOF)
{
    avg = (float) g1 / g2;
    printf("avg = %.2f", avg);
    status = fscanf(in, "%d%d", &g1, &g2);
}
```

لو الملف فارغ او غير موجود يرجع EOF



Pointers := dynamic addresses

العنوان 5

```c
int x=5, z=30;
int *y = &x;
```

int ← نوع Pointer

في الذاكرة: لكل شيء مكان وعنوان

X [5] (100) → adress

printf("%d", &x); → X→100

يطبع الـ address ولكن ليس القيمة

و هنا يطبعها 100

→ y

X [5]

(y) → 100

```c
y = &z;
```

• ولو كتبت هيك تصير y تؤشر على Z

• يعني لا تتغير

• الفرق X تبقى ثابتة لو حطيتها ولكن لا تتغير

مثال

y → Z [7] 200

• يطبع 200

What is the output?

```c
int x=3; y=2;
int *a=&x; *b;
printf("%d\n", a);      ①
printf("%d\n", &x);
b=&y;                   ②
printf("%d\n", b);
a=b;                    ③
printf("%d\n, *b);
*a=20;                  ④
printf("%d\n", y);
```

ما بتأشر على حدا (*b)

يطبع 100

يطبع 100

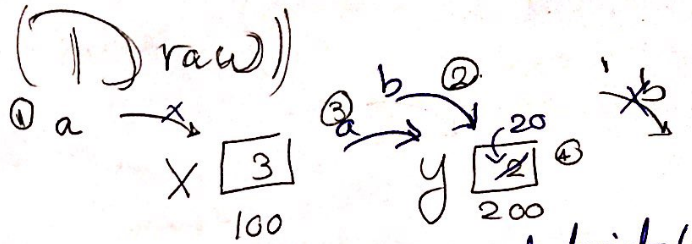
أصبحت تؤشر على y

يطبع 200

صحيح لأنه a و b نفس النوع

- تختلف عن int *
- تسمى تحته indirection
- يعني أعطيني ما تؤشر عليه b
- يعني تطبع 2
- & يعني ال address

(Draw)



• what it prints (answer)

100
100
200
2
20

Notes:-

a = x; • خطأ لأنه a و x ليسا من نفس النوع (int)

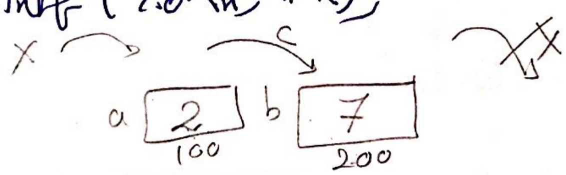
a = &x • صحيح لان الاثنين pointers

What is the output

```c
int a=2, b=7;
int *c=&b, *x;
printf(%d\n", *c);
*c=a+2;
printf("%d\n", b);
x=&a;
*x=*x+5;
printf("%d\n", a);
printf("%d\n, x);
```

a=b+a

printf("%d\n", *x);



7
4
7
100
7
7

lecture 20

نزيد عدد الـ Pointers

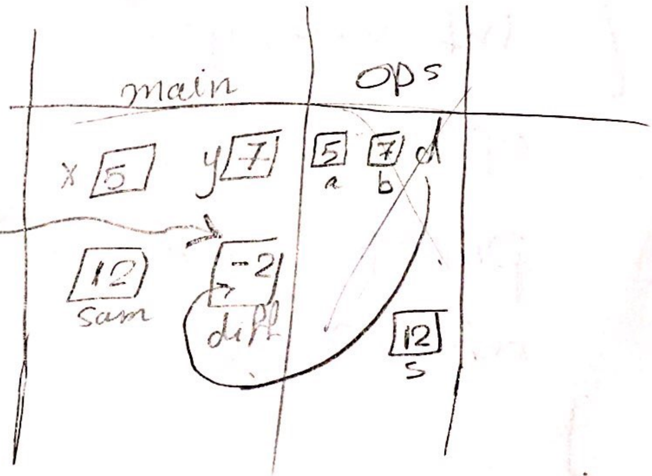
- function ترجع شيئ واحد int*
  لكن تستطيع ارجاع اكثر من شيئ عن طريق الـ Pointers

```
#include <stdio.h>
int ops( int, int, int *);      ← output parameters
int main ( )
{
    int x=5 y=7, sum, diff;
    ops (x, y, &sum, &diff)
    sum = ops(x, y, &diff);
    printf ("sum=%d" diff =%d", sum, diff);
    return 0;
}
```

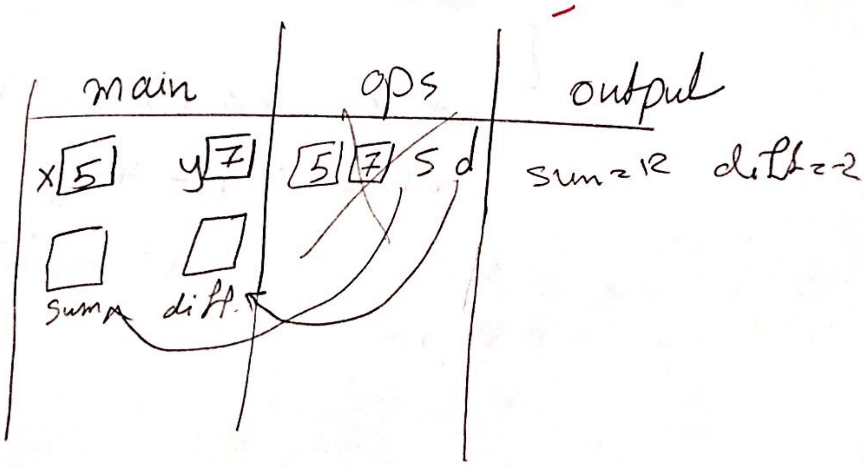
(argument → x, y)

- لكي ترجع اكثر من شيئ (بالرصاص)

```
void int ops(int a, int b, int *d), int *s)      int a = x;
                                                 int *d = &diff;
{
    int s;
    s = a+b;
    *d = a-b;
    return s;
}
```

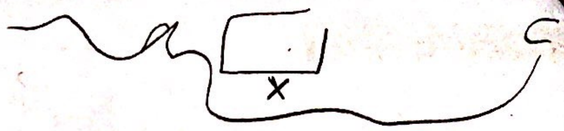


الغلبة كتالك



Scan f :-

int scanf (——, int * c)



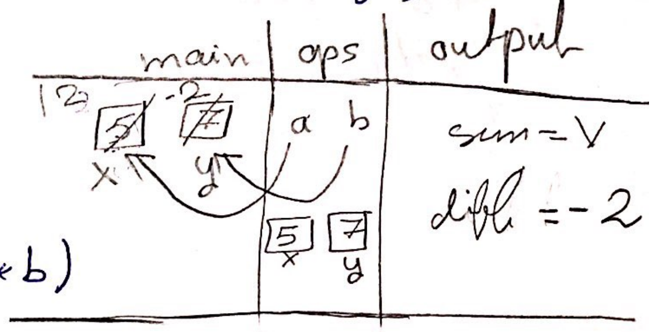
• اذا ما حطيت & بيدخل ال Pointer لـ Trash ، يعني لا يوخذ ما ادخل المربع بل يذهب الى مكان غير معروف في الذاكرة

• مع & : يؤشر على المربع واذا حطيت 5 بتحطها في المربع

• There is a Void * C (Void pointer) علامة المستقبل البعيد

```
#include <stdio.h>
Void ops (int*, int *);
  int main (   )
{ int x=5, y=7;
  ops (&x, &y);
  printf ("sum=%d  diff=%d ", x, y);
  return 0;
}
```

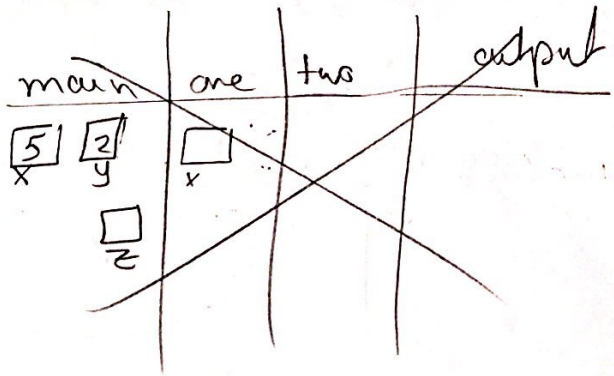
Important

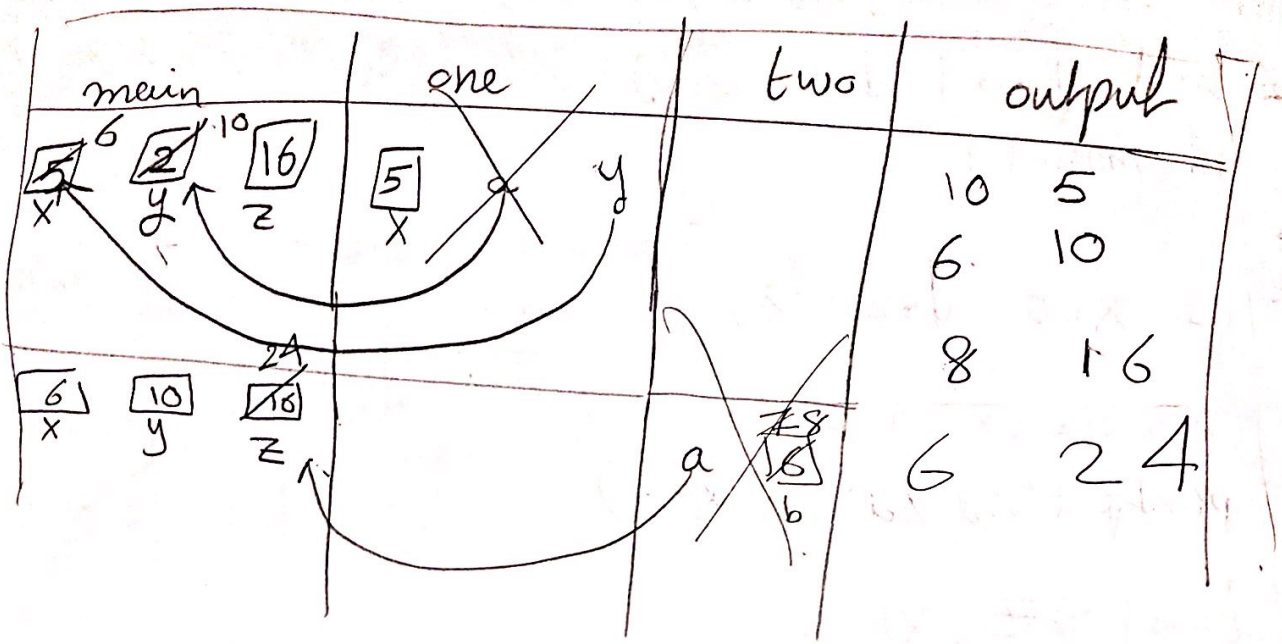



```
Void ops (int *a, int*b)
{
  *a = *a + *b ; wrong
  *b = *a - *b;
  int x= *a , y = *b;
  * a = x+y;
  * b = x-y;
}
```

What is the output?

```
#include <stdio.h>
int one( int, int *, int *);
void two (int *, int);
int main ()
{
  int x=5, y=2, z;
  z= one (x, &y, &x);
  printf ("%d %d\n", x, y);
  two (&z, x);
  printf ("%d %d", x, z);
  return 0;
}
int one (int x, int *a, int *y)
{
  *a = x+*y;
  printf ("%d %d\n", *a, x);
  (*y)++;
  return *a + *y;
}
void two (int *a, int *b)
{
  b++;
  printf ("%d %d", ++b, *a);
  *a = *a+b;
```





lecture 20

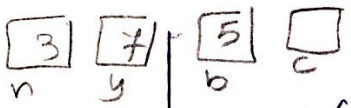
* global vs local variabiles

```
#include <stdio.h>
int x=5;          → global variabl
int one(int);
int main()
{
  int y
  y=x;
  :
}
int one(int b)
{ int x          →
  x=10;
  :
}
```

- لو في متغير اسمو نفس اسم الـglobal (X) مثلاً فانه بيطغى عليه ويستغل هو
- لو ما كان في، بستخدم الـglobal
- لو int x ما كانت موجودة تغير قيمة الـglobal وتصير 10 مش 5

---

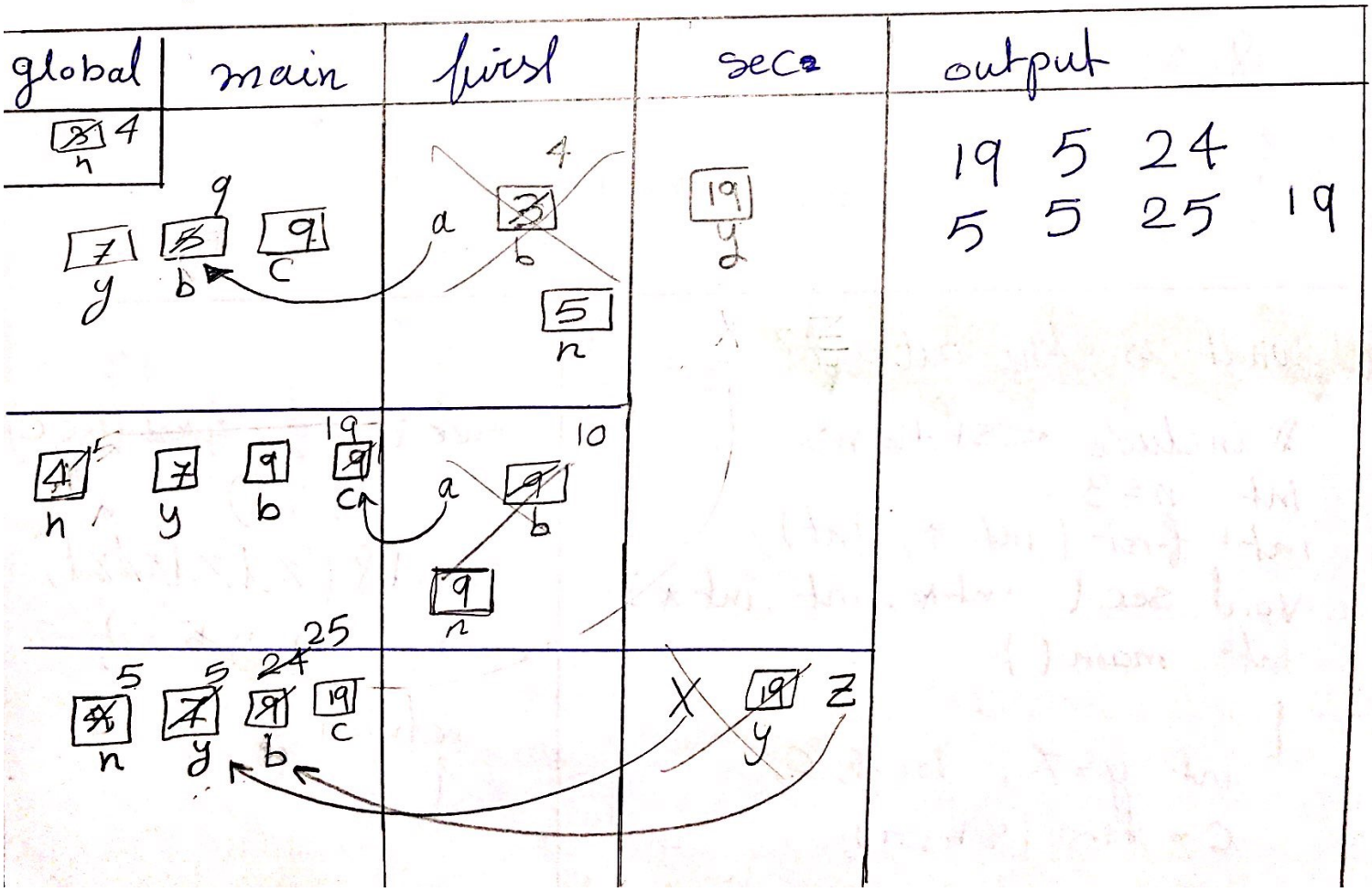
what is the output?




```
#include <stdio.h>
int n=3;
int first(int *, int);
void sec(int*, int, int*);
int main()
{
  int y=7, b=5, c;
  c=first(&b, n);
  sec(&y, first(&c, c)
      , &b);
  printf("%d%d%d%d",
      n, y, b, c);
  return 0;
}
```

19

```
int first (int *a, int b)
{
 int n = *a;
 b++;
 *a = n+b;
 return b+n;
}

Void sec ( int *x, int y, int*z)
{
   *x = ++n;
   *z = *x+y;
  printf ("%d %d %d\n", y, n, *z);
  (*z)++;
}
```



Pointers :-
↳ Dynamic address

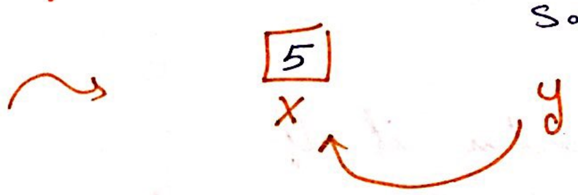
• Notes :-
if a is a pointer and x is a variable Then you can't say x=a but you can say a = &x

→ How to define it ?

```
int *y = &x;
```

int ← نوع المؤشر

&x ← الموقع الذي يؤشر عليه



→ What is the Output ?

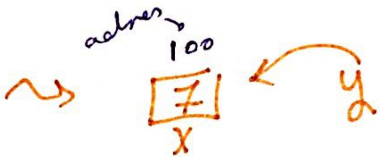
If you have a pointer *y :

→ printf("%d", y); → it prints the address ~~they~~ that y is pointing to it.

→ Printf ("%d", *y); → it prints the value inside the address

Ex

```
int x=7;
*y = &x;
```




```
Printf ("% d", y);
Printf ("% d",*y);
```

output

100
7

→ How to use pointers in functions ?

• Remember that a function ~~is~~ can return one value only But we can use pointers to return more than one value from a function.

How?

first

The function should be void / Example: function to calculate sum/multipl division & substr

```
void ops (int, int, int *, int *, int *, int *)
```

You can add As many As you want

Second :-

When Calling the function :-

ops (x, y, & sum, & multi, & div, & sub)

Third :-

In the function itself :-

```
ops (int a , int b, int * d , int * S , int *F, int *E)
 {
   *d = a+b ;
   *S = a*b ;
   * F = a/b ;
   * E = a-b ;
 }
```

Note

When it comes to printing when you have many functions : let's say we have 2 functions : Sum and Sub

```
: Z = Sum ( x, y, s---- ) ;
Printf ("%d\n", Z, x) ;
|
E = Sub ( N, F, ---- ) ;
Printf ("%d%d", N, F ---- ) ;
```

You work out Z if there is another Print in the function it comes first

Then This

same

Lastely this

Global vs local variables

- If you define a variable before main function. it's a Global variable

Ex
```
#include ___
int x=5
|
|
int main( )
```

- in functions, If there is a variable same as a global variable then :-

```
int one (int b)
{
  int x
  x=10;
```

→ You take this Value
But if it didn't say it's value then you take X=5

• لو كانت مش موجودة
تصبح قيمة global عشرة وليس خمسة

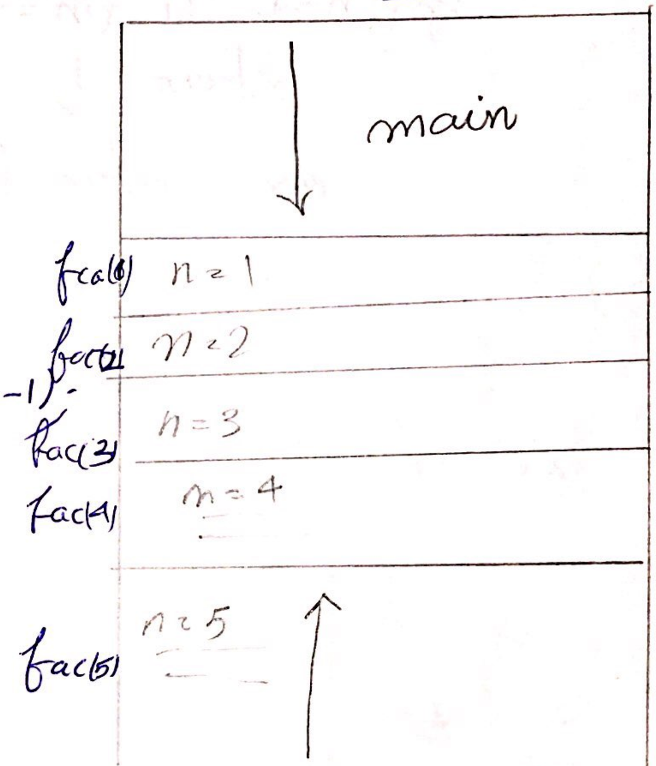
lecture 26

Recursion

fac(5) = 5 * 4 * 3 * 2 * 1
= 5 * fac(4)
= 4 * fac(3)
3 * fac(2)
2 fac(1)

```
int fac (int n)
{
  if(n == 1)
    return 1;
  else
    return n * fac(n-1);
}
```

stack: مكدسة
بتحط من فوق
مش من تحت
LIFO
last in first out




```
int unknown(int x, int y)
{
  if (y == 1)
    return x;
  else
    return x + unknown (x, y-1);
}
```

unknown (3, 4) → 12
return 3 + unknow(3, 3) → 9
3 + unknow(3, 2) → 6
3 + unknown (3, 1) → 3
ret 3

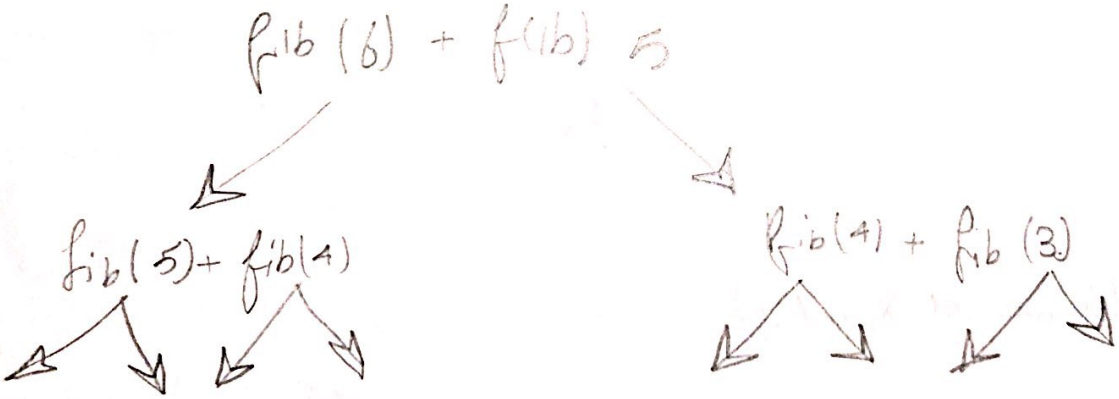
fibonacci :-

	1	1	2	3	5	8	13	21 - - - - - - -
n=	1	2	3	4	5	6	7	8 - - - - - - - -

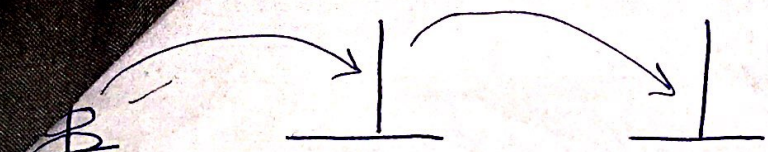
```
int fib(int n)
{
  if((n==1) || (n==2))
      return 1;
  else return (fib(n-1) + fib(n-2));
}
```

• هذا البرنامج يعطي جواب ، لكن يأخذ وقت طويل

Ex:- fib(7)



• الـ Recursion بتاخد loop



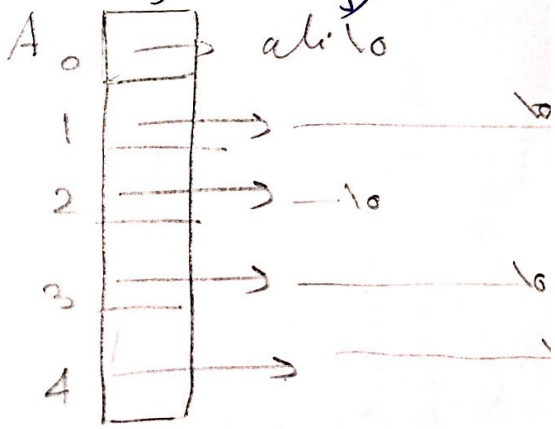
5234 16

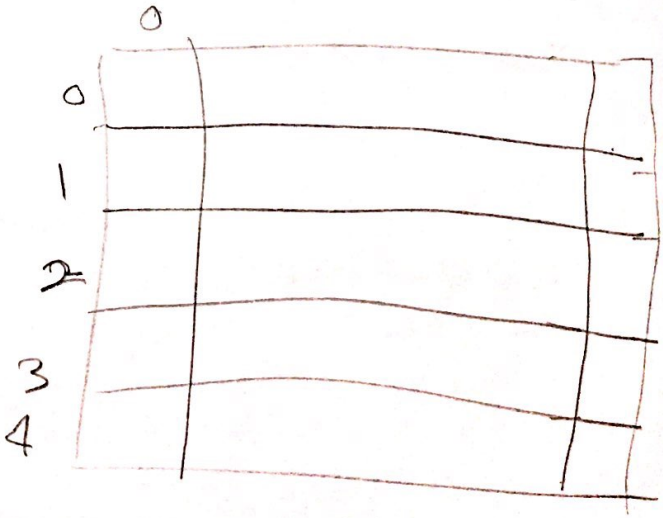
to print

5 2 3 4 16

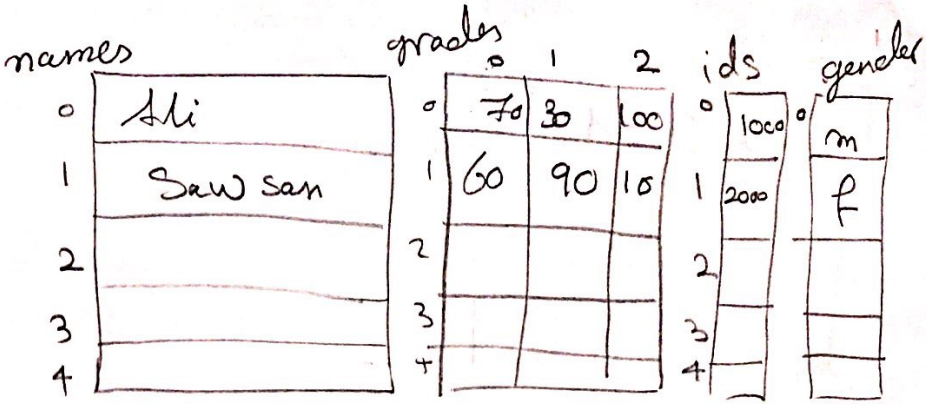
- array of pointers

Char * A[5] = {"ali", "Ahmad", — };





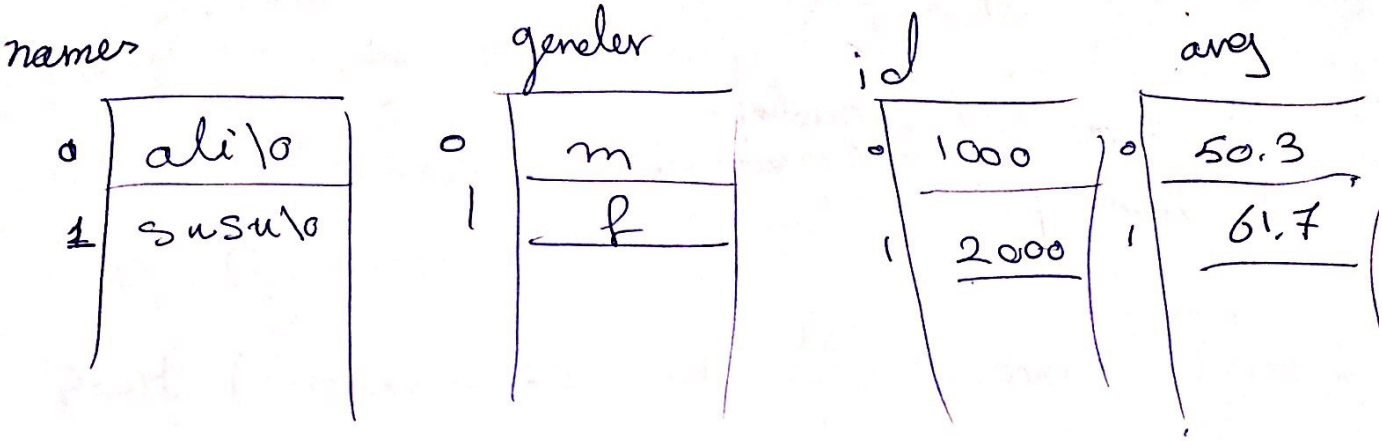
• Parallel Arrays :-



avg (float)

Lecture 27

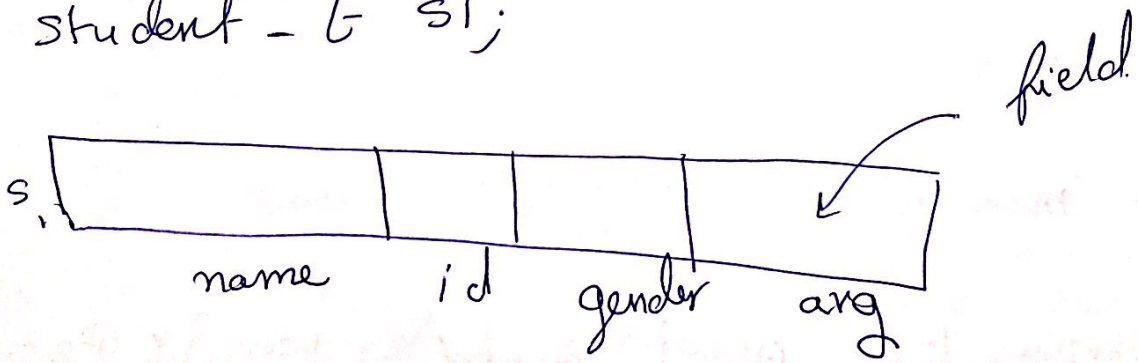
Structures :-



Typedef Struct

```
{
    char   name[10];
    int    id;
    char   gender;
    double avg;
}
    student_t;
```

```
student_t s1;
```




```
strcpy ( s1.name , "ahmad");
s1.id = 1000;                    s1.avg = 70.7;
s1.gender = 'm';
```

Array of structures:

student_t students[10];

students

0	name	id	gender	-
1	name	id		

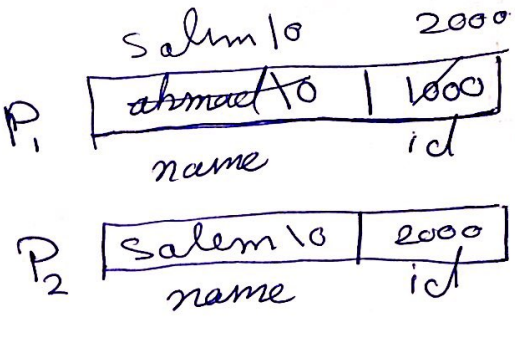
~ . ~ . ~ . ~ . ~ . ~ . ~ .

```
#include <stdio.h>
#define S 10
typedef struct
{
    char name[S];
    int id;
}
  person_t;

  X  y;

int main()
{
  person_t  p1 = {"ahmad", 1000}; p2;
```

p2 = p1; ← You can do that

if (p1 == p2) X You can't do that

```
printf ("Enter name & id \n");
scanf ("%s%d", p2.name, &p2.id);
p1 = p2;
printf ("name = %s id%d", p1.name, p1.id);
return 0;
}
```



output

Enter na——.

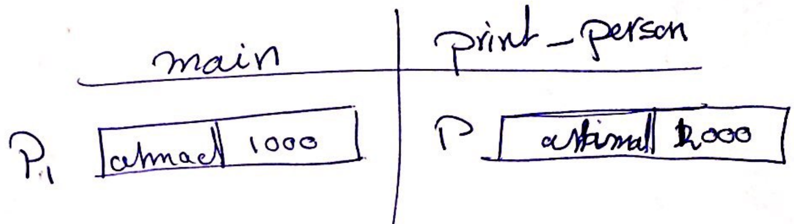
salem 2000

Using a function

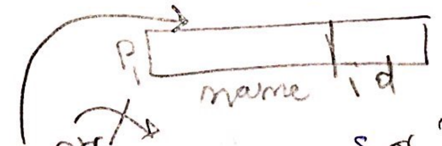
| same definition

```
Void print_person (person_t);
int main ()
{
```

```
person_t  p1 = {"ahmad", 1000};
print_person (p1);
return 0;
}
void print_person ( person_t  p)
{
    printf (" name = %s   id =%d", p.name, p.id);
}
```



output
name = ahmad
id = 1000

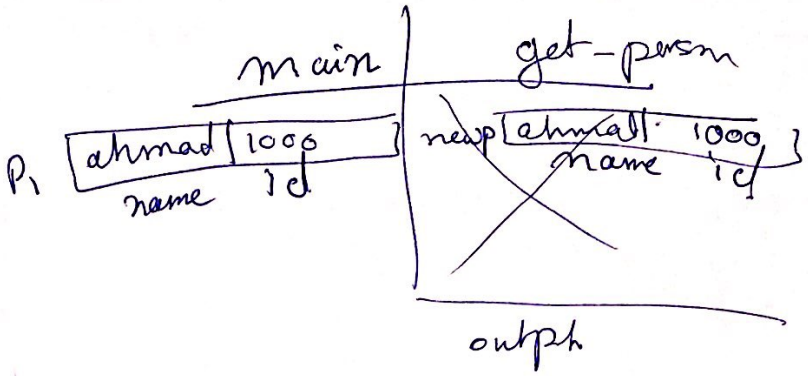



```
#define S 10
typedef struct
{
    char name [S];
    int id;
} person_t;
person_t get_person ( );
int main ()
{
    person_t p1;
    p_ptr = &p1
    p1 = get_person ();
    return 0;
}
person_t get_person ()
{
    person_t newp;
    printf ("Enter name & id \n");
```

(*p_ptr).name { or p-ptr -> name

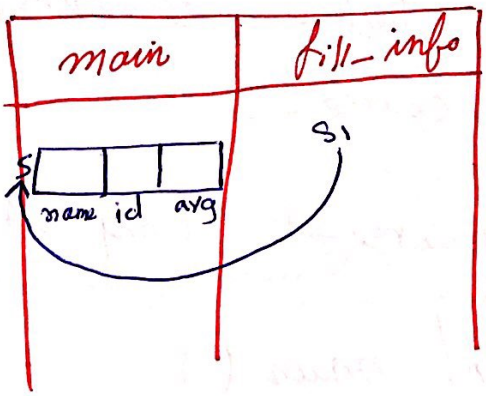
*p-ptr;

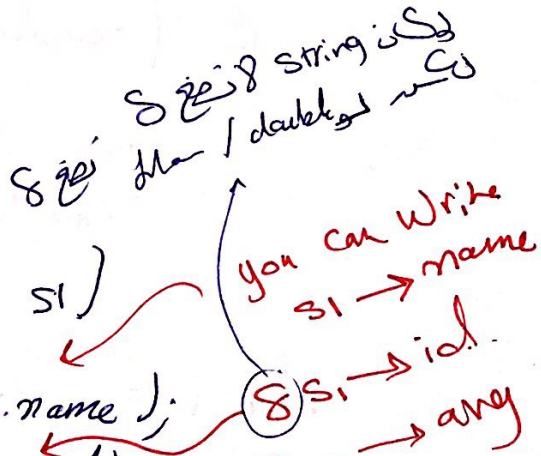
```
nf ("%s %d", newp.name, &newp.id);

return newp;
}
```



Enter name & id
ahmad 1000

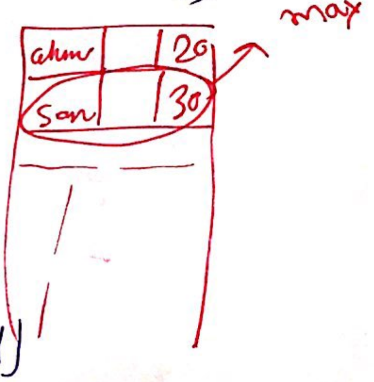
```
#include <stdio.h>
#define S 10
typedefr struct
{
    char name[S];
    int id;
    double avg;
}
Void fill_info (stud_t);
int main ()
{
    stud_t S;
    printf ("Enter name, id and avg for student\n");
    fill_info (&S);
    printf ("%s %d %.2f", S.name, S.id, S.avg);
    return 0;
}
Void fill_info (student * S1)
{
    scanf ("%s", (*S1).name);
    scanf ("%d", &(*S1).id);
    scanf ("%lf", &(*S1).avg);
}
```





```
typedef struct
{
    char instructor [5];
    int num ;
    stud_t students [20];
} course_t

course_t comp124;
```

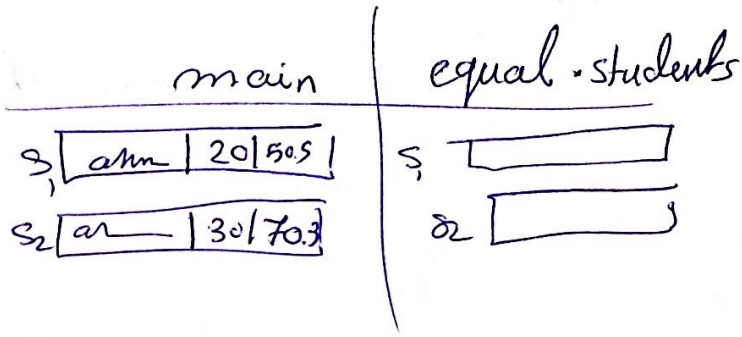
```
int main ( )
{
    stud_t students [10], max_stude , temp;
    int i ;
    for ( i=0; i <10 ; i++)
        fill info ( & student [i] ); → or
    [ { students[i] .ma
        scanf("%s", )
        scanf("%d", &stud [i].id
        scan
        ,
        ,
    } ]

    max_stden = students [0];
    for (i=0; i<10; i++)
        if ( students[i] .avg > max_student.avg )
            max_student = students [i];
            printf (" nam = %s", max_stu.name);

    for (i=0; i<5-1; i++)
        for (j=0; j<5-1; j++)
            if ( student [j].id > student [j+1].id)
```



```
temp = Students[j];
Students[j] = Students[j+1];
Students[j+1] = temp;
}
sum=0, count=0;
for (i=0; i<10; i++)
{
if (strcmp(students[i].name, "ahmad")==0)
{
sum += students[i].avg;
count++;
}
avg_ahmad = (float) sum/count;
```

```
#include <stdio.h>
int equal_students(student_t, student_t);
int main()
{
student_t S1 = {"ahmad", 20, 50.5};
S2 = {"ahmad", 30, ~~30.~~ 74.3};
if (equal_student (S1, S2))
printf("same\n");
else
printf("diff\n");
return 0;
```

```
int f equal_Students ( Stud-t s1 , Stud-t s2 );
{
    return (strcmp( s1.name , s2.name) == 0
            && s1.id == s2.id
            && s1.avg == s2.avg)
}
```



lecture 29 :-

- **Files :-**

```
FILE  *in;
in = Fopen ("data.txt" , "r");
   if ( in == NULL)
   {
        printf ("Cannot open file");
        exit (0);
   }
```

exit(0) ← يعني كلشي عام

exit( ) ← لتقدر تحط أي رقم

```
FILE *in;
Char   filename [10];
printf ("Enter fill name \n");
scanf ("%s", filename);
in = fopen (file name , "r");
if (in == NULL)
 {
   Printf("Cannot open file %s", file name);
   exit (2);
 }
```

exit(2) ← هذا الرقم يعني انه هناك Erro

Enter file name
- data ←
الاسم غلط لانه الموجود في السيستم هو data.txt
لذلك
- Cannot open
- Enter file
الجواب هون NULL يعني ع أحد الطرف
يظل يدخل في Loop ويرجع

```
FILE *in ;
Char filename [10];
Printf ("Enter file name\n");
for ( scanf ("%s", filename); (in = fopen(filename, "r")) == NULL
      ; scanf ("%s", filename))
```

{ يطبع هذه ويطبع التي تحتها ويطلب اسم ملف آخر
printf ("Cannot open file %s ", filename);
printf ("Re enter file name\n"); 8 تساوي NULL
}

output (again)

Ent
data
Connect
Re enter file
dataa.txt
Canne
Re
data.txt

hi.exe

disadvantage:
1- Nor readable
2- differant between systems

Binary files:-

← disadvantages of text files:-
1- processing time
2- precision problem في حالة الـ float تكون هناك أرقام ناقصة بسبب التقريب
3- space

```
FILE *out;
int x=3456, y=74;
fprintf (out, "%d\n%d", x, y);
fscanf (in, "%d%d", &x, &y);
```

Memory chars (ملاحظة)

x: 010 ... → (3456) 10
y: 01 ...

هذه كلها تأخذ الرقم (3465) هذه char
↓ تحولها إلى
decimal
↓
(3465)

File
3456 74
5.27

```
fprintf (out, "%.2f", x);
```
وكان لدينا double

How to open Binary files?

```
FILE *in;
in = fopen ("data.bin", "rb");
fread (—, —, —, —);
fwrite (—, —, —, —);
fclose (in);
```

"rb" → read binary

"wb" → write binary

بتكتب نوع أي حجم

- الاختلاف يكون في القراءة والكتابة

↳ للقراءة نستخدم fread

↳ للكتابة نستخدم fwrite

كيف أكتب وأقرأ من file

```
int x=7, y; float a=3.5, b;
FILE * out;
Out = fopen("data.bin", "wb");
fwrite (&x, sizeof (int), 1, out);
```

- عنوانه x
- اقصى حجم int من الذاكرة memo
  على الشاشة

```
fwrite (&a, sizeof (float), 1, out);
fclose (out);
out = fopen ("data.bin", "rb");
fread (&y, sizeof (int), 1, out);
fread (&b, sizeof (float), 1, out);
printf ("X=%d  b=%.2f", x, b);
```

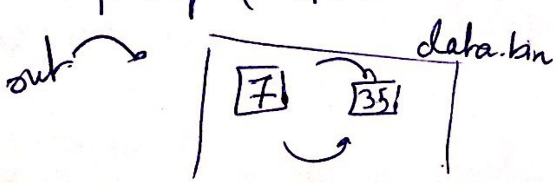
8-bit → 2byte

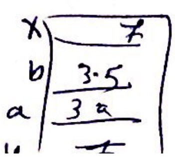
4 byte → float

int n=sizeof (int); → 2

sizeof (float); → 4

sizeof (X) → 2





X=7 , b=3.5

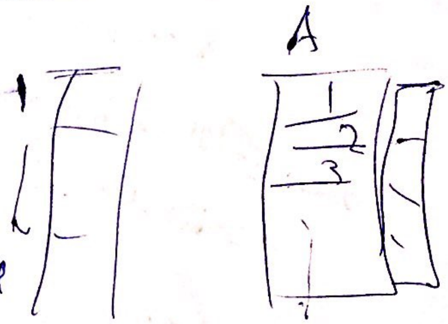
```c
typedef struct
{
    char name [10];   → 4byte
    int id;           → 2byte
} stud_t;

stud_t s1 = {"ahmad", 10};
fprintf (out, "%s %d", s1.name, s1.id);
fscanf (out, "%s %d" myname, &my id);
fwrite (&s1, sizeof (stud_t), 1, out);
fread (&s2, size of (stud_t), 1, out);
```

أكبر من الفايل وحط B على out آخر ←

لما أعرف Array في الذاكرة لازم يكون مع نص

(عكس write read)

```c
int B[10];
int A[5] = { 1,2,3,4,5 };
fwrite (A, sizeof(int), 5, out);
fread (B, sizeof(int, 5 out);
fread (&B[5], sizeof (int), 5, out);
```



70

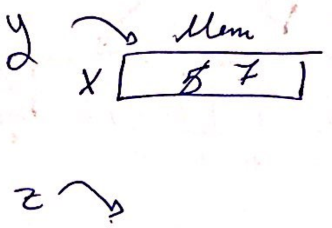
```c
int n;
n = fread (A, sizeof (int), Looo, out);
```

↗ ترجع كم صحيح قرأ من File

lecture 30

Dynamic allocation :-

```
int x=5;
int *y = &x;
int * z;
*y = 7; ✓
* z = 3; ✗
int A[1000];
int *A ;    int A[ ];
```



• malloz (memory allocate) ← موجودة داخل #include <stdlib.h>

```
int   *x;
float *x
X = (int * ) malloc( size of(int)) ;
```

تنفع لأي نوع

int بتاخذ الشيء العام من malloc وحوّله لـ

void pointer

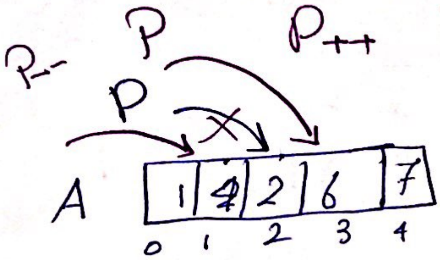
```
X = (int *) malloz (sizeof (int) * 10)
```

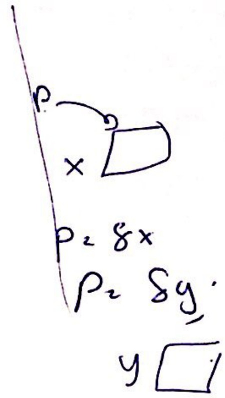
يحجز 10 int

CalloC بتستخدم لحجز مستمر

```
X = (int *) Calloc (10, sizeof(int));
```

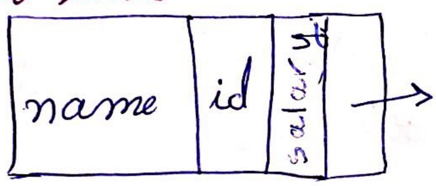
```
int A[5] = {1, 4, 2, 6, 7};

int *p;
```

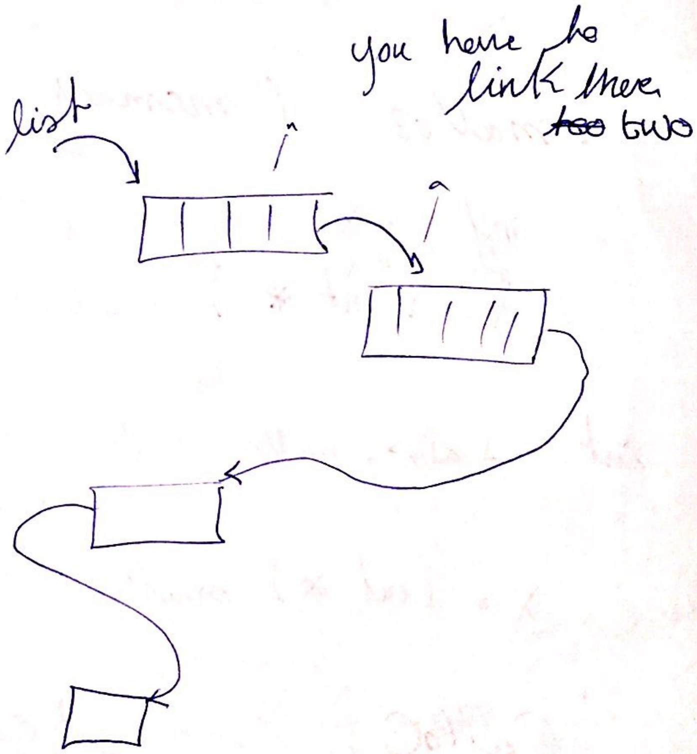





```
P = A;
p = &A[0];
P = &A[2];
```

- structure ال
linked list:






```
typedef struct stud_a
{
    char name[10];
    int id;
    struct stud_a *next;
} stud_t;
```

```
int main ( )
{ Char name[10]. int id;
  Stud_t *list;
Printf ("Enter name and id\n");
Scanf ("%s %d", name, &id);
list = (Stud_t * ) malloc (sizeof (Stud_t));
```

strcpy (list → name, name);

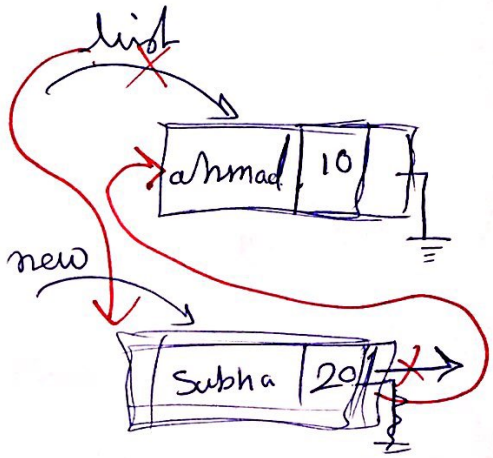
```
list → id = id;
list → next = NULL;
```

You define new like list

```
new → next = list.
```

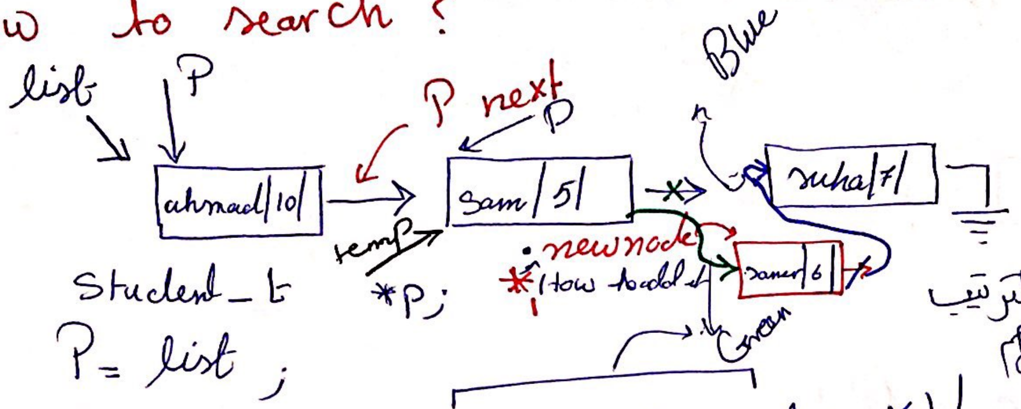
linked

```
list = new;
```



lecture 31

How to search?



• لازم يكون بهاد الترتيب
ممنوع ان تعكسهم

```
Student_t *p;
P = list;

while ((p != NULL) && (p->id != x))
    p = p->next;
if (p != NULL)
printf("name = %s", p->name);
    else
printf("%d No such id", x);
```

u have to add this → if (p != NULL)

How to add?

```
*1 :- newnode = (stud_t *) malloc(sizeof(stud_t));
      newnode->next = p->next;   This step (Blue)
      p->next = newnode;          This step (Green)
```

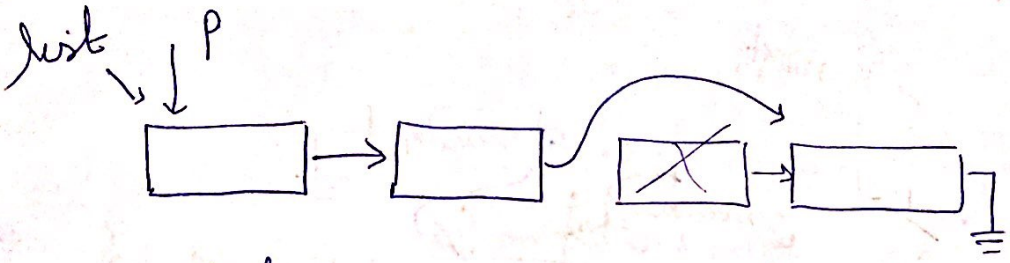
How to delete?

```
temp = p->next;
p->next = temp->next;
free(temp);
```

(or you can say :- p->next = p->next->next;)

free(temp); ← function عكس الـ malloc

How to print the list ?




```
p = list;
while ( p != NULL)
{
    printf ("%s\n", p->name);
    printf ("%d\n", p->id);
    p = p->next;
}
```

```
Char temp [20];
Char name [10][20];
char *name [10];
for (i=0; i<10; i++)
{
    Printf ("Enter name \n");
    Scanf ("%s", temp);
    names[i] = (char *)malloc
        (sizeof (char) * (strlen(temp)+1));
    Strcpy (names[i], tmp);
}
```

names