codeblocks.org

```
#include <stdio.h>   Standard input/output header file
int main
{
Data Type ← int num1, num2;   → variables
          int sum;
printf("Enter first number\n");
scanf("%d", &num1);
printf("Enter second number\n");
scanf("%d", &num2);
sum = num1 + num2;   → Assignment Statement
printf("sum = %d", sum);
return 0;
}
```

يطبع كل اللي بالقوسين حاجة اللي بتبدأ بـ % أو \

int
float
double
char

%d → decimal

& → عنوان

Enter ⇦ ؟

```
printf("Enter any two numbers\n");
scanf("%d %d", &num1, &num2);
```

Enter any two numbers ↵ { Enter... ... ↵ 5 ␣ 6 ↵ } → screen

لو دخل 5 ↵ ثم 6

Comments:

/* ——— */

in C++ //

data types:

int	printf	%d	scanf	%d
float	printf	%f	←	%f
double	printf	%f	scanf	%Lf
char	printf	%c	scanf	%c

```c
# include <stdio.h>
int main ()
{
    int age ;
    char gender;
    Printf ("Enter age and gender\n") ;
    scanf ("%d %c" , &age, &gender) ;
    printf ("age= %d gender = %c", age , gender ) ;
    return 0 ;
}
```

screen:

Enter age and gender
20 F ↵

age = 20 gender = ___ (char هي space لأنو)

عشان نتجنب هاي المشكلة بنحط

```c
scanf ("%d ␣ %c", &age ; &gender );
```

بنحط space عشان
تطنش لـ space char

scanf تنس

"%d %d %f ␣ %c %d"

Arithmetic Operation :

+ `x = y + 2;`

− `x = y - 3;`

* `a = b * c;`

/ → `b = 5/2 = 2` (integer) — قسمة للأعداد الصحيحة

% → `b = 5%2 = 1` (integer باقي) — باقي القسمة

`3%7 = 3`

`5+2*7/3-2` ← الأولويات

والأقواس تغير الأولويات →

```
#include <stdio.h>
int main()
{
    int num, ones, tens, hunds, rev;
    Printf("Enter any three digit number \n");
    Scanf("%d", &num);
    ones = num%10;
    hunds = num/100;
    tens = num/10%10;        // tens = num%100/10;
    rev = ones*100 + tens*10 + hund;
    Printf(" reverse of %d = %d ", num, rev);
    return 0;
}
```

برنامج يحسب مساحة ومحيط الدائرة

```c
#include <stdio.h>
#define PI 3.14
```

(بتفصل space) → عرفنا PI كثابت

```c
int main ()
{
    int rad ;
    float area , circum;
    Printf (" Enter radius \n");
    Scanf ("%d", & rad );
    area = PI * rad * rad;
    circum = 2 * PI * rad ;
    Printf ("Area = %f \n", area );
    Printf (" circum= %f ", circum);
    return 0;
}
```

نوعها float لأن Area عدد عشري ~ π

→ `printf("Area =%f \n circum = %f" ;`

إذا ما حددت خط كم خانات بعد الفاصلة بطلع الجواب مثلاً : Area= 27.500000

Type Casting :

```c
float x;
Int y ;
```

y = 10 / 3 ; → 3

x = 10 / 3 ; → 3 .0 (ناتج عدد صحيح صحيح)

لو حطينا :

y = 10.0 / 3 ; → 3

x = 10.0 / 3 → 3.3

```
int x = 5 , y = 2 ;
float b = 7.3 ;
int res ;

float fres ;

res = x / y ;  → 2

fres = x / y ;  → 2.0

fres = (float) x / y ;  → 2.5
     = x / (float) y ;
     = (float) x / (float) y ;

res = b % x ;
      ↑
    error
   (int) b  بنحط

b = x ;
float  int
```

→ عادي نحط int بداخل float
لأنو المتغير بالكبير

```
x = b ;   ✗
x = (int) b ;  ✓
```


Error Types:

1. syntax errors + warnings
2. Run Time (linker) errors (fatel)
3. logical errors : sum = num1 * num2

x = 5;
y = 5 - x;
z = 3/y;

~~3/0~~ ✗ 3/0

Formatting Output:

int :

X = 5237 , y = 2 , z = 12345;

```
printf ("%d %d %d", x, y, z);
```

5237 2 12345

3 27 125 ← لو وعدل وغير القيم

26 74 62 30 123456

⇐ بدنا إياهم مرتبين

عشان موجب بجي من اليمين

```
⇒ printf ("%10d %7d %6d", x, y, z);
```

_ _ _ _ _ _ 5 2 3 7 _ _ _ _ _ _ 2 _ _ 1 2 3 4 5

↑ عشان space

%-2d لو حطينا عدد خانات غير كافي زي

5 2 . 3 7 _ _ _ _ _ _ _

↑ الكومبايلر بيجيبه وبيكمل عادي

char : char b = 'A' ;

```
printf ("%7d %6c %-3d", x, b, z);
```

5 2 3 7 _ _ _ | _ _ _ _ _ A | _ | 1 2 3 4 5

↑ space

x = -5237 ;

-5237 _ _

```
float x = 25.2764 , y = -3.245 , z = 2 ;

printf ("%-10.3f  %7.2f %06.4f", x, y, z);
```
تقريب أول 3 خانات :

25.276_ _ _ _ | | _ _ -3.24 2.0000

"%0.3f" → ما في width
بس بدو 3 خانات بعد الفاصلة

```
int x = -32775;
char y = 'B' ;
float z = -627.37496 ;
printf ("%-10.4f %03c %9d", z, y, x);
```

-627.3750_ _ _ _B_ _ _ _ -32775

---

a = 9/6 = 3/2 = 1
b = 1
k = 1.0
m = 1.5

```
printf ("this % %%")
```
% زي ما هي نطبعها

Text Files:

```
#include <stdio.h>
int main ()
{
    int num1, num2 ,sum;
    FILE *in ;
    in = fopen ("data.txt", "r");
    fscanf (in, "%d%d", &num1, &num2);
    sum = num1 + num2 ;
    printf ("sum=%d", sum);
    fclose (in) ;
    return 0;
}
```

pointer (*in) – نقدر نغير اسمها

FILE هاي نوع in

mode: r : قراءة, w : كتابة

fclose (in): بتفك العلاقة بين in و data.txt

in → data.txt

data.txt	
5	6

```c
#include <stdio.h>
main()
{
  int num1, num2, sum;
  FILE *in, *out;
  in = fopen("date.txt", "r");
  out = fopen("result.txt", "w");
  fscanf(in, "%d%d", &num1, &num2);
  sum = num1 + num2;
  fprintf(out, "sum=%d", sum);
  fclose(in);
  fclose(out);
  return 0;
}
```

Function:

system defined functions:

```c
#include <math.h>
pow(x, y);   // x^y
sqrt(x);     // √x
abs(x);      // integer absolute value
fabs(x);     // double absolute value
ceil(x);     // smallest integer >= x   (1.2) -> 2
floor(x);    // largest integer <= x    (1.2) -> 1
sin(x);      // x in radians
cos(x);
tan(x);
```

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

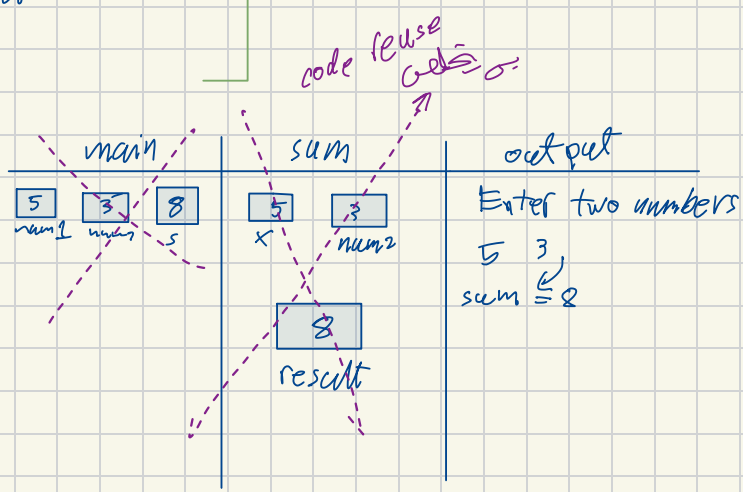
```c
z = sqrt(pow(x,2) + pow(y,2));   // x*x , y*y
```

```c
ceil(-1.4);   // -> -1
floor(-1.4);  // -> -2
```

user defined function:

```
#include <stdio.h>
int sum (int, int);          → function prototype
int main()
{
    int num1, num2, s;
    printf ("Enter two numbers\n");
    scanf ("%d %d", &num1, &num2");
    s= sum(num1, num2);      → function call
    printf ("sum = %d", s);
    return 0;
}

int sum (int x, int num2)
{
    int result;
    result = x + num2;
    return result
}
```

→ function definition



```
#include <stdio.h>
void sum (int, int);
int main ()
{
    int num1, num2, s;
    printf ("Enter two numbers\n");
    scanf ("%d %d", &num1, &num2);
    sum (num1, num2);
    return 0;
}

Void sum (int x, int num2);
{
    int result;
    result = x + num2;
    printf ("sum = %d", result)
```

```
#include <stdio.h>
int sum ()          ;
int main ()
{
    int s;
    s = sum();
    printf ("sum = %d", s);
    retur 0;
}

int sum ();
{
    int num1, num2, s;
    printf("enter two numbers\n");
    scanf("%d %d", &num1, &num2);
    s = num1 + num2;
    return s;
}
```

```
#include <stdio.h>
void sum ( );
int main ( )
{
    sum ( );
    return 0;
}

void sum ( )
{
 int num1, num2, s;
 printf("enter two numbers\n");
 scanf("%d %d", &num1, &num2);
 s = num1 + num2;
 printf ("sum = %d", s);
}
```

main	sum	output
	5 (num1) 3 (num2) 8 (s)	Enter two numbers 5 3 sum = 8

$y = \overset{a}{3}x^{\overset{3}{}} - \overset{b}{2}x^{\overset{2}{}} + \overset{c}{x} - \overset{d}{5}$

find-y

cube

squar

```
#include <stdio.h>
int find_y (int, int, int, int, int);
int cube (int);
int sqr (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 = find_y (a, b, c, d, x);
  printf ("y = %d", y);
  return 0;
}
int sqr (int x);
{
  return x*x;
}
```

الحل

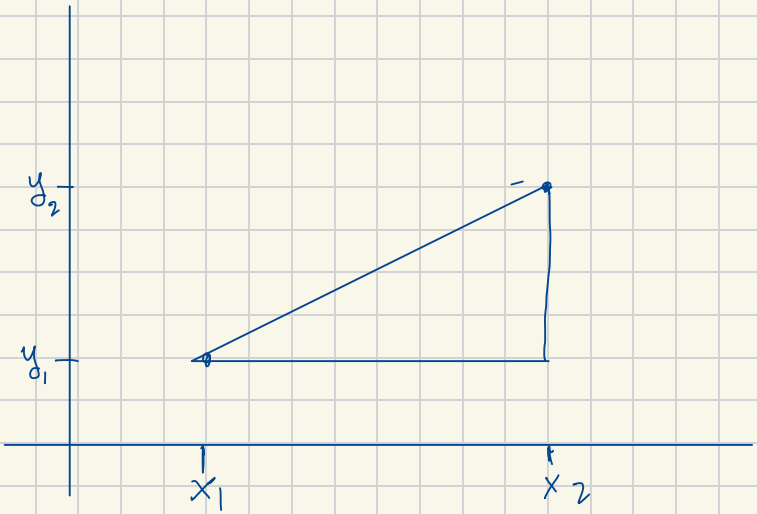
```
int res
res = x*x
return res
```

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

→ return x * sqr(x) الحل

```
int find_y (int a, int b, int c, int d, int x);
{
  return a*cube(x) + b*sqr(x) + c*x + d;
}
```

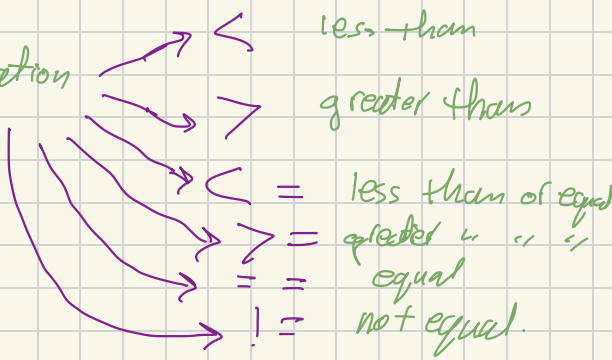
File
New
No
all files



Selections :

```
x = 2;
If (x > 5)
    printf ("hi\n");
else
    printf ("bye\n");
```

relational operation



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

Good طلع

x = 0
if (x = 0) → bad

false الصفر يعتبر

true غير

if (5 % 2) → 1 طلع Good بدل true

if (x) : true غير False غير الصفر يعتبر !

```
int x
printf (" Enter value\n");
scanf ("%d", &x);
If (x >= 0)
    printf ("%d is positive", x);
Else
    printf ("%d is negative", x);
```

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

```
if (x % 2)  او  if (result)
    printf ("odd");
else
    printf ("even");
```

```c
int mark
printf ("enter mark 0-100 \n");
scanf ("%d", &mark);
if ( mark >= 90)
{
    printf ("grade = A \n");
    printf ("good job \n");
}
else if ( mark >= 80)
    printf ("grade = B");
else if ( mark >= 70)
    printf ("grade = C");
else
{   printf ("grade = F");
    printf (" good luck next time\n");
}
printf ("goodbye");
```

compound { (two statements → 1, 2)

logical operators

! not

&& and

|| or

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

إذا فش أقواس الأولوية ! بعدها && بعدها ||

```c
int avg = 85;
char gender = 'f';
int age = 22;
if ( (avg > 80) || (age < 30) && ! ( gender == 'f'))
```

(T || T && ! (T))

(T || T && F)

(T || F) ⟹ T

if(((avg > 80) || (age < 30)) && ! (gender == 'f'))
(T || T) && ! (T)
(T && F)
F

---

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

if((mark >= 70) && (mark < 80))
  printf("grade = C");
if((mark >= 80) && (mark < 90))
  printf("grade = B");

---

if (x=5) : ; ← if

---

int x=2, y=7, z=10;
max = x;
if(y > max)
    max = y;
if(z > max)
    max = z;
printf("max = %d", max);

---

char letter;
printf("Enter letter\n");
scanf(" %c", &letter);
if (letter

letter == 'a' || 'e' || 'i' || 'o' || 'u'

letter == 'a' || letter == 'o' : true

Boolean functions: (true or false) 1 0

```
# include <stdio.h>
int isEven (int)
int main ()
{
    int num, result;
    printf ("enter number\n");
    scanf ("%d", &num);
    result = isEven (num);
    if (result == 0)
        printf ("%d is not even", num);
    else
        printf ("%d is even", num);
    return 0;
}

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

return !(n%2)

if (isEven(num) == 0)

if (isEven(num)

is even

else

not even

Switch Statement

```c
int x =1;
if (x==1)
   printf ("one\n");
if else (x==2)
   printf ("two\n";
if else (x==3)
   printf (" three "\n);
else
   printf (" unknown number\n");
```

using switch (في حالة الـ أو)

```c
int x = 1;
switch (x)
{
   case 1: printf ("one\n");
           break;
   case 2: printf ("two\n"); break;
   case 3: printf ("three\n"); break;
   default : printf (" unknown num\n");
}
```

اذا لقى break بطّل ينزل

لو شلنا هاي الـ break

لو x = 1 بطبع 1 و 2

لو x = 2 بطبع 2

switch بس للـ int و char

```c
char letter;
if ( letter=='a' || letter =='o' || letter ----- )
          vowel
else
          not-vowel
```

```
switch (letter)
{ case 'a': printf ("vouel"); X
            break;

case 'a': case 'u': case 'e': case i: case 'o': printf ("Voul");
                                                 break;
 default: printf ("Not a vocal");
}
```

---

menu

```
int n1, n2;
int result;
int choice
printf("enter any two numbers \n");
scanf ("%d%d", &n1, &n2);
printf("select one of the following 1-3 : \n");
printf(1- Add\n 2- subtract\n 3- Multiply\n");
 scanf("%d", &choice);
switch (choic)
{
   case1: result = n1 + n2; break;
   cas2 2: "     = n1 - n2; "  ;
   case3: "     = n1 * n2; "  ;
   default: printf(" No such choice\n");
}printf ("result = %d", result);
```

Enter ----
5 6
select ----
1- ~~Add~~ Add
2- subtract
3 - Multiply
2

Nested If:

x=5 , y=2 , z=4;

```
if (x>y)
    if (y==z)
        printf("good\n");
else
    printf("bad");
    printf("bye");
```

x=5 y=2 → good bye

y=2 x=1 → bye

x=5 y=3 → bad bye

```
else
{
    printf("bad");
    printf("bye");
}
```

x=5 y=2 → good

y=2 x=1 → (Nothing)

x=5 y=3 → bad bye

```
if (x>y)
{
    if (y==z)
        printf("good\n");
}
else
```

x=5 y=2 → Good

y=2 x=1 → bad bye

x=5 y=3 → bye

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

x=5 y=2 → good

x=1 y=2 → bad bye

x=5 y=3 → —

Loops:

while
for
do/while

```
x=5 → initial value
while (x<=10) → final value (condition)
{
    printf("%d\n", x);
    x++; // x=x+1;
}
```

change ← y

```
x++;  post increment
++x;  pre increment
x--;  post decrement
--x;  pre decrement
```

```
x=5;
printf("%d\n",x);
x++;
printf("%d\n",x);
```

5	5
x++	++x
6	6

(لو كانت في سطر لحالها)

```
x=5, y=2;
printf("%d\n",x);
y= x++
printf("%d\n",x);
printf("%d\n",y);
```

بالنسبة لـ x

5	5
x++	++x
6	6
5	6

بالنسبة لـ y

بيقرأ الأصلية

```
x=2;
printf("%d", x++); → 2
          ++x); → 3
```

عكل ما تلقى زي واحد

```
x=2;
if(++x > 2) →
```

بتزيد x++ لو

---

x = x+5 →	x+=5
x = x-3 →	x-=3
x = x/y →	x/=y
x = x%2 →	x%=2

```
y = 1+2+3+4 .... +20
y=0;
i=1;
while(i<=20)
{ y+=i; // y=y+i;
  i++;
}
```

x^y $2^4 \rightarrow 16$

مثال: $y = 4$ $res = 2 * 2 * 2 * 2$

```
res = 1;
i = 1;
while (i <= y)
{
    res *= x;
    i++;
}
```

---

n = 20 , sum of devisors :

sum = 1 + 2 + 4 + 5 + 10 + 20

```
sum = 0
i = 1;
while (i <= n)
{
    if (n % i == 0)
        sum = sum + i;
    i++;
}
```

For :

```
for (initial value ; condition ; counter)
{
    // statment2
    // statment2
}
```

(1) (2) (3) (2) ... يضل يلف

Ex:

```
for (int i = 0 ; i < 5 ; i++)
{
    printf ("%d\n", i);
}
```

0
1
2
3
4

```
for (int i=1; i<10 ; i++)
{
  printf("%d",i);
}
```

1 2 3 4 5 6 7 8 9

---

program to print even number between 0-100

```
for (int i=0; i<=100; i=i+2)
{
  printf("%d",i);
}
```

---

the sum of the first 10 numbers

```
int sum=0;
for (int i=1; i<=10 ; i++)
{
   sum = sum + i;
}
printf("the sum is %d", sum);
```

the sum is 55

```
int i, j, rows;
printf("enter number of rows");
Scanf("%d", &rows);
for (i=1; i<=rows; i++)
{ for(j=1; j<=i; j++)
  { printf("*");
  }
```

Do While:

statement;
do
{
  statement;
}
while (condition);

Ex:
```
int i=0;
while (i==1)
{
  printf("hi");
  i++
}
```

```
int i=0;
do
{
  printf("hi");
  i++;
}
while (i==1);
```

Ex:

```
int num, sum=0;
do
{
  printf("enter integer or 0 to stop");
  scanf("%d", &num);
  if (num>0)
    sum += num;
}
while (num!=0);
```

بداية مقارنة التغيير

```
for (i=1; i<5; i++)
    printf("%d", i);
```

```
for (i=1, j=5; ((i<4) && (j>2)); i++, j--)
    printf("i=%d  j=%d", i, j);
```

```
num=1364;
count=0;
while(num>0)
{
  count++;
  num=num/10;
}
printf("count=%d", count);
```

(4)

⟸ صاحي ما بتزبط لو num=0 بالبداية

```
num=1364;
count=0;
do {
   count++;
   num=num/10;
} while(num>0);
```

do while
بلف لفة وحدة على الأقل

```
int x=1;
while (x<=7)
{
    printf("x = %d \n", x);
    if (x==5)
        break; => يطلع من اللوب
    printf ("x = %d \n", x);
    x++;
}
printf ("bye");
return 0;
}
```

يطبع الأرقام من 1 لـ 4 مرتين
و 5 يطبعها مرة

```
int x=1;
while (x<=7)
{
    printf("x = %d \n", x++); → 1   بأول لفة
    if (x==6)
        break; => يطلع من اللوب
    printf ("x = %d \n", ++x); → 3
    // x++;
}
printf ("bye");
return 0;
}
```

بأول لفة يطبع 1 بعدها 3

1
3
5
5
5

```c
int x=1;
while (x<=7)
{
    printf("x = %d \n", x);
    if (x==6)
        continue;
    printf ("x = %d \n", x);
    x++;
}
printf("bye");
return 0;
}
```

continue; ⇐ بيرجع ع الشرط مرة ثانية

ما بيكمل اللفة

infinite loop

رح تضلها تلف لأنو x = 6 (بيضلها)

```c
int x=1;
while (x<=7)
{
    printf("x = %d \n", x++);
    if (x==6)
        continue;
    printf ("x = %d \n", x);

}
printf("bye");
return 0;
}
```

إعادة دراسة

1
2
2
3
3
4
4
5
5
6
7
7
8

STUDENTS-HUB.com Uploaded By: anonymous

Ex:

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

بدنا نفصل إشي زي هيك ⟸

بس بـ loop

Nested loop:

program that finds if a given number n is prime or not

أولي (يقسم ع حالو و 1 بس)

```c
int num;
printf("Enter any number\n");
scanf("%d", &num);
int i=2;
int prime=1;          // flage  → prime Lotion true
while (i < num)
{
    if (num % i == 0)
    {
        prime = 0;
        break;
    }
    i++;
}
if (prime)
    printf("%d is prime", num);
else
    printf("%d is not prime", num);
return 0;
```

→

```c
if (num % i == 0)
    printf("Not prime\n");
else
    printf("prime");
i++,
```

هون غلط لأنو رح يطبعلي يحكي
من prime مرة مش prime

1000 تک 1 کے prime کی

```
int num, i, prime;
for (num=1; num<=1000; num++)
{
  i=2;
  prime=1;
  while (i<num)
  {
    if (num%i==0)
    {
      prime=0;
      break;
    }
    i++;
  }
  if (prime)
    printf("%d is prime", num);
}
  return 0;
```

---

fn. =

```
int main
{
int isprime(int);
int num;
printf("Enter any number\n");
scanf("%d", &num);
if (isprime(num))
    printf("%d is prime\n", num);
else
    printf("%d is not prime\n", num);
return 0;
}
```

```
int isprime (int n)
{
    int i=2;
    while (i<n)
    { if (n% i == 0)
          return 0;
       i++;
    }
    return 1;
}
```

بدنا prime من 1 لـ 100

نفس قبل بس بإضافة :

```
int num;
for (num=1; num<=100; num++)
    if (isprime)
```

Nested loops :

one Dimension

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

col

```
      * * *
raws  * * *
      * * *
```

عدد

3

```
مرة 3   for (i=1, i<=n; i++)
         {for (j=1; j<=n; j++)
3 مرة <=      printf ("*");
         printf ("\n");
3 x 3 = 9 }
```

هاي رح تلف 9 مرات

* * * * * * * * *

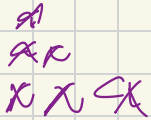
هاي بطبع

\n ته

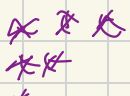
```
[ * * *
  * * *
  * * * ]
```

```
i = 1 ;
while (i <= n)
{
    j = 1 ;
    while (j <= n)
    {
        ...
        j++ ;
    }
    i++ ;
}
```

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



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



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

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

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

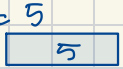
```
for ( i=1, i<=n, i++)
{
    for (j=i; j<n; j++)
        printf (" ");
    for ( k=1; k<=i; k++)
        printf ("%d", k);
    printf (" \n");
}
```

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

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

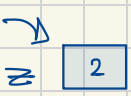
Pointers. dynamic address

```
x = 5
```



x ← static address = 1000

```
y = &x;
```

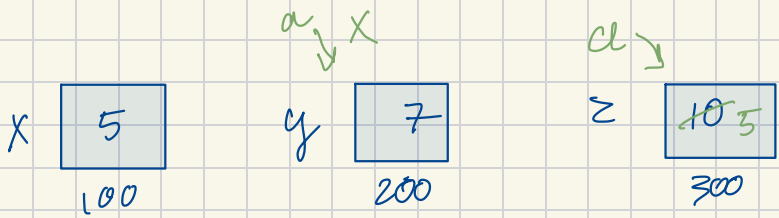



```
printf ("x = %d", *y);
```

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

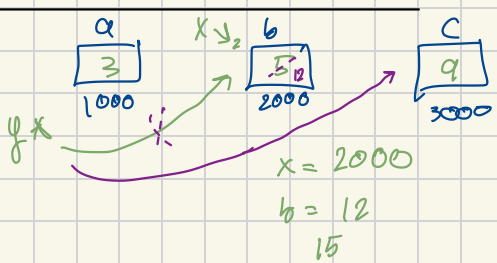
```
int *y = &x ;
```

```
int z = 7 ;
    y = &z ;
```



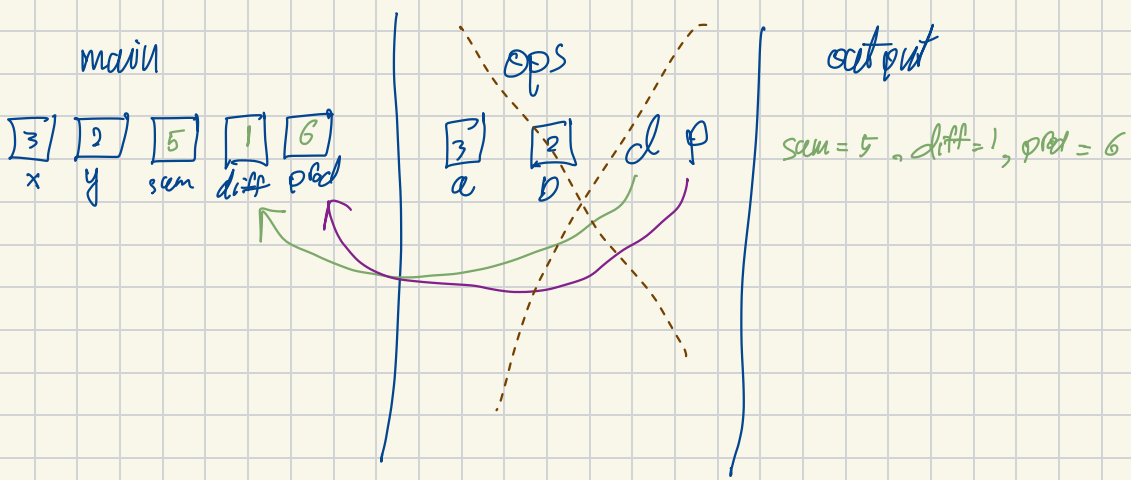

```
int *a = &y ;
printf ("%d", a) ;
a = &z ;
printf ("%d", *a) ;
*a = x ;
printf ("%d\n", z) ;
```

```
int a=3, b=5, c=a ;
int *x = &b, *y ;
printf ("x = %d\n", x) ;
y = x ;
*y = 12 ;
printf ("b = %d\n", b) ;
y = &c ;
c = a+b ;
printf ("%d\n", *y) ;
```

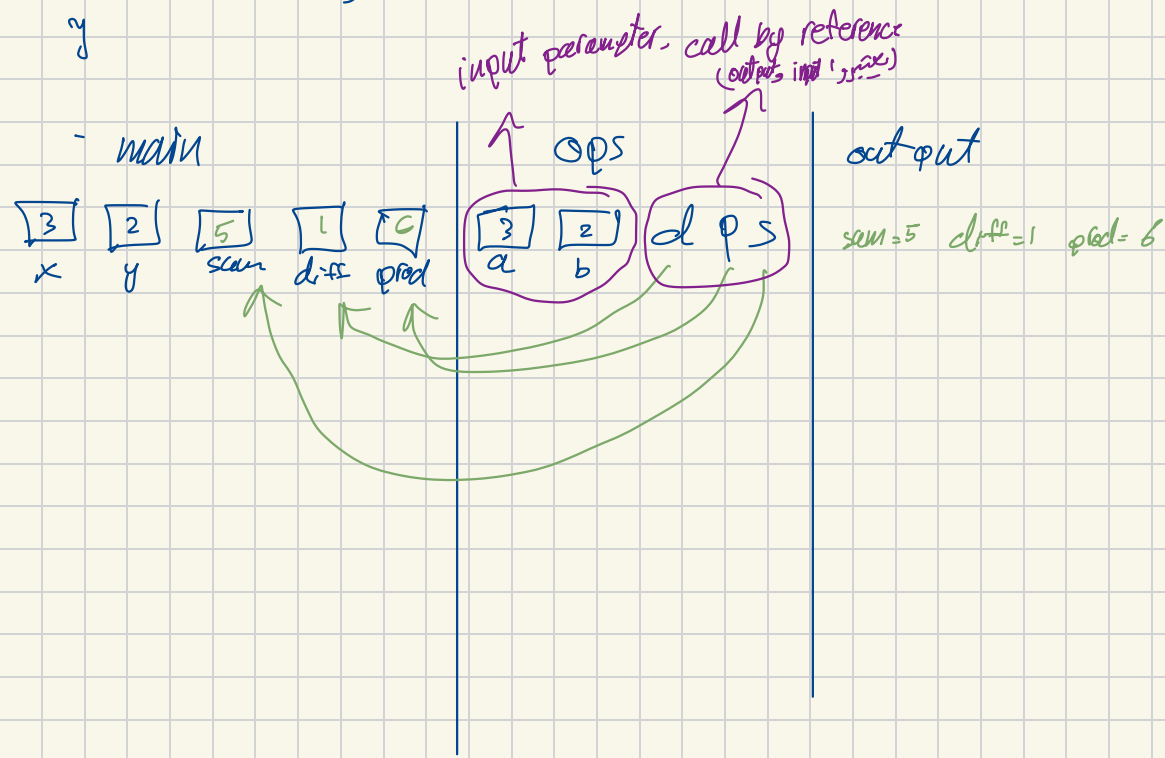


```
#include <stdio.h>
int ops(int, int, int*, int*);
int main()
{
    int x=3, y=2;
    int sum, diff, prod;
    sum = ops(x, y, &diff, &prod);
    printf("sum = %d, diff = %d, prod = %d", sum, diff, prod);
    return 0;
}

int ops(int a, int b, int *d, int *p)
{
    *d = a-b;
    *p = a*b;
    return d+p;
}
```

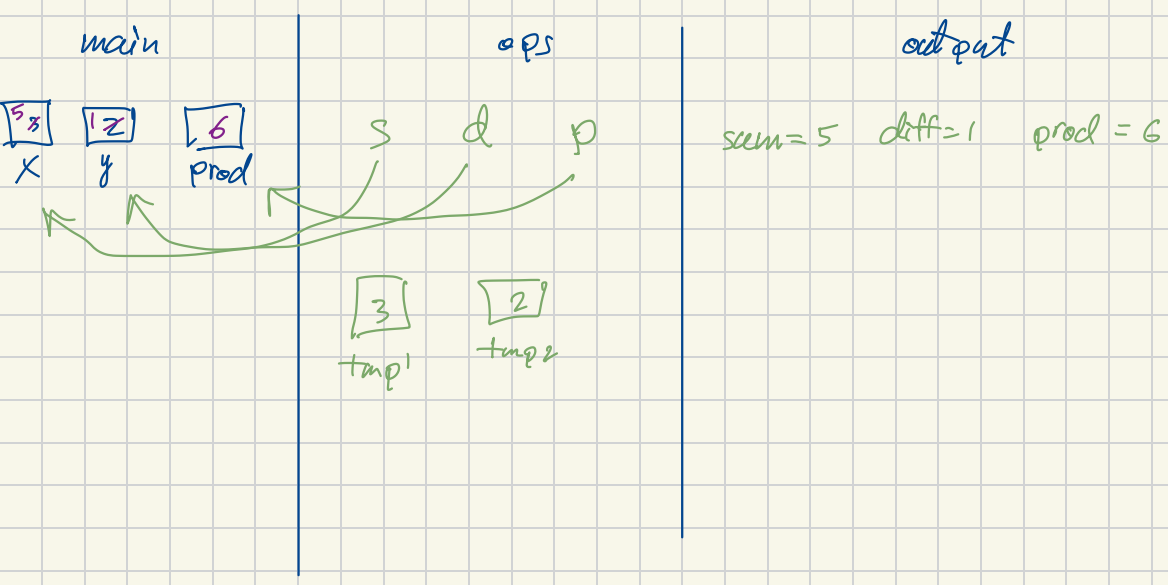
int *d = &diff



```
#include <stdio.h>
void ops (int , int , int* , int*);
int main()
{
   int x=3, y=2;
   int sum , diff , prod;
       ops (x,y,& diff, & prod , & sum);
   printf("sum = %d , diff=%d, prod = %d" , sum , diff, prod);
   return 0;
}

void ops (int a, int b , int *d, int * p ,int * s)
{
    *d = a-b ;
    *p = a*b ;
    * s = a+b ;
}
```



```
#include <stdio.h>
int ops(int *, int*, int*);
int main()
{
   int x=3, y=2;
   int prod;
   ops(&x, &y, &prod);
   printf("sum = %d, diff = %d, prod = %d", x, y, prod);
   return 0;
}

void ops(int *s, int *d, int *p)
{
   int tmp1 = *s, tmp2 = *d;
   *s = tmp1 + tmp2;
   *d = tmp1 - tmp2;
   *p = tmp1 * tmp2;
}
```

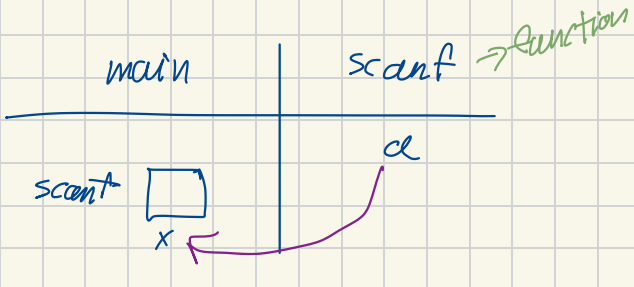


```
int x;
scanf ("%d", &x);
```

scanf(, int *a) → scanf(, void *a)

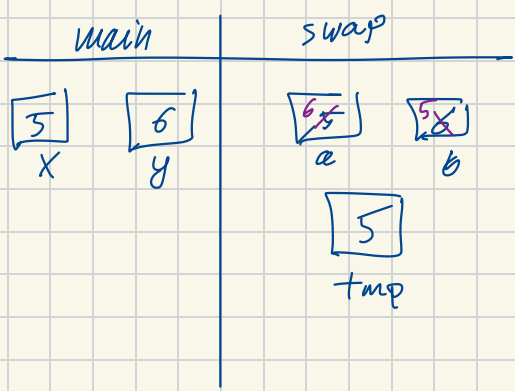
int → 4 bytes

void → 0 bytes



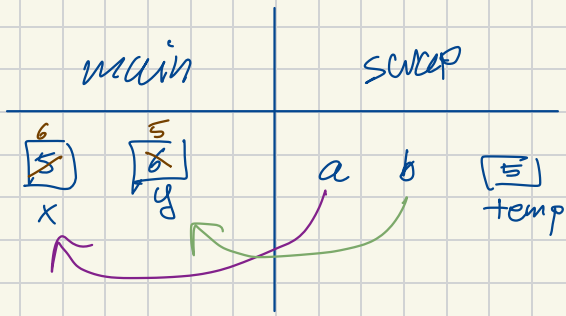

```
#include <stdio.h>
void swap(int, int);
int main()
{
    int x=5, y=6;
    swap(x, y);
    printf("x=%d  y=%d", x, y);
    return 0;
}

void swap(int a, int b)
{
    int tmp;
    tmp = a;
    a = b;
    b = tmp;
}
```

```
x=5, y=6;
int temp;
temp = x;
x = y;
y = temp;
```



```
#include <stdio.h>
void swap(int*, int*);
int main()
{
    int x=5, y=6;
    swap(&x,&y);
    printf("x=%d  y=%d", x, y);
    return 0;
}
```

```
void swap(int*a, int*b)
{
  int tmp;
  tmp = *a;
  *a = *b;
  *b = tmp;
}
```



Global variables

local variables

scope rule

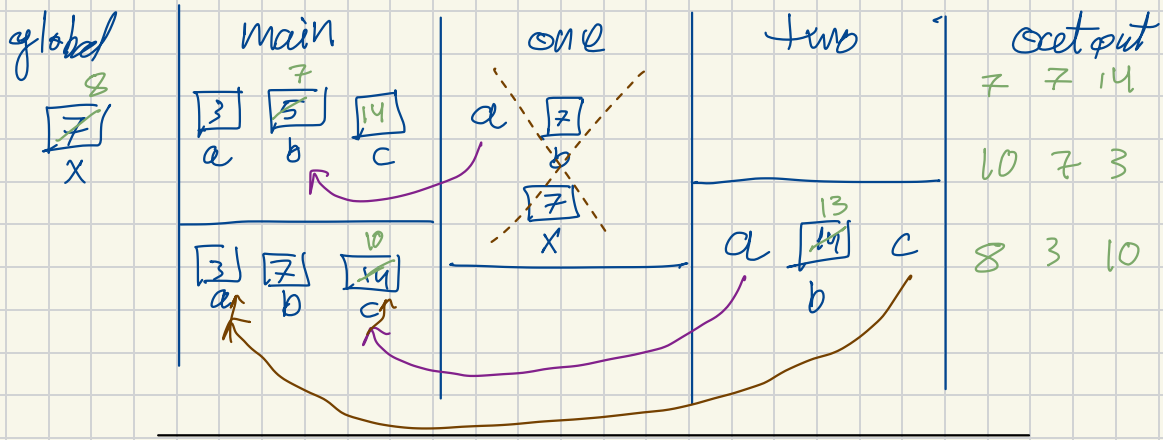
```
int main()
{
    int a=3;
}
```
local main

```
int fun(int x)
{
    int a=7, b=2;
}
```
local fun

```
#include <stdio.h>
int n=2;
int main()
{
    int a=3;
    int n=10;
    printf(..., n);  → 10
}
```

```
int fun(int x)
{
    int a=7, b=2;
    n++;
    printf(..., n);  → 3
}
```

```
#include <stdio.h>
int x = 7;
int one (int *, int);
void two (int *, int, int *);

int main()
{
    int a=3, b=5, c;

    c= one (&b, x);
    printf("%d %d %d\n", x, b, c);
    two (&c, c, &a);
    printf("%d %d %d", x, a, c);
    return 0;
}

int one (int *a, int b)
{
    int x = b;
    *a = x;
    return x + *a;
}

void two (int *a, int b, int *c)
{
    *a = *c + x;
    b = *a + 3;
    printf("%d %d %d\n", *a, x, *c);
    x++;
}
```



Q: Write a C program that sends the length and width of a rectangle to a function called vals and return both, the area and circumference of the rectangle

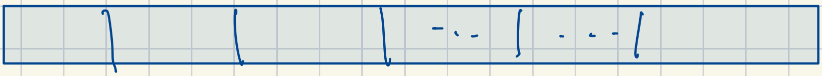
```
#include <stdio.h>
int vals ( int , int , int * );

int main()
  {
    int length , width , area , circumf;
    area = int vals ( length, width, &circum)

    printf (" area = %od circum = %od ", area, circum);
    return ;
  }

int vals ( int L, int w , int *c)
  {
    *c = 2*L + 2*w;
    return L*w;
  }
```

Arrays

تخزين علامات 100 طالب من نوع int

```
int stuM[100]
```

نوع — اسم — حجم

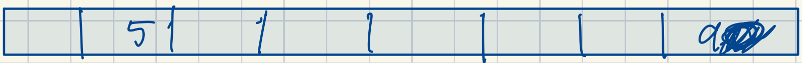


stuM[0] stuM[1] ---[2] ---- ---- stuM[99]

(index ↑ stuM[0])

```
int x [8]
```

Array x



x[0] x[7]

```
x[7] = 9
x[1] = 5
printf ("%d", x[5]);      5
int sum
sum = x[1] + x[7]         14
```

Array initialization

```
            y[0] y[1] y[2] y[3]
int y [5] = {71, 73, 79, 83, 89};
int i = 3;
printf ("%d %d", i, y[i]);      →  83
printf ("%d", y[i] + 1);        →  84
 "     ("%d", y[i+1]);          →  89
 "     ("%d", y[i++]);          →  83
y [i - 1] = y[i]     3 = i  →
                     79 يطبع
```

```
int s[] = { 3, 5, 4, 7, 50, 6 };
printf("%d", s[4]);  → 50
                s[10] → garbage
```

```
int square[11], i;

for (i=0; i<11; i++)
  {
    square[i] = i * i;
    printf("%d\n", square[i]);
  }
```

0 1 4 9 16 25 100

```
int i, sum=0, arr[10];
printf("enter 10 numbers");
for (i=0; i<10; i++)
  {
    scanf("%d", &arr[i]);
    sum=
```

```
int i, count=0, arr[10];
for(i=0; i<10; i++)
  {
    scanf("%d", &arr[i]);
    if arr[i] == 1)
      count += 1
  }
  printf("%d", count);
```

```
int i, a[5];
printf("enter 5 numbers");
for (i=0; i<5; i++)
{
  printf("element %d", i);
  scanf("%d", &a[i]);
}
printf("the values reverse");
for (i=4; i>=0; i--)
{ printf("%d", i);
}
```

```
int max(int arr[], int n);
int main
{
  int a[5];
  printf("enter 5 numbers");
  for (i=0, i<5, i++)
  {
    printf("element %d", i)
    scanf("%d", &a[i])
  }
  printf("the max num is %d", max(a,5));
}
int max(int arr[], int n)
{
  int max, int i;
  for (i=0; i<n, i++)
  {
    if (max < arr[i])
      max = arr[i];
  }
  return max;
}
```

max = arr[0] first is 11

11 7 9 12 10

Finding array size

sizeof( ) → array کے ہر خانے کا سائز

```
int main()
{
    int numbers[5] = {10, 20, 30, 40, 50};
    int size = sizeof(numbers);
    printf("%d", size);
}
```

int → 4 bytes

→ 20 (5×4)

پوری array کا سائز → sizeof(x) / sizeof(x[0])

---

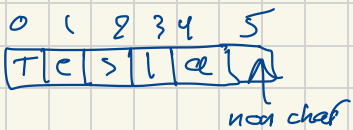
string → "Ali"

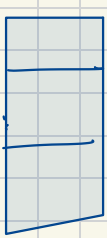
جو لفظ ڈبل کوٹیشن میں ہو

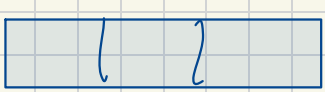
Declaring string arrays

```
int main()
{
    char car[6] = "Tesla";
    printf("%c", car[2]);
}
```

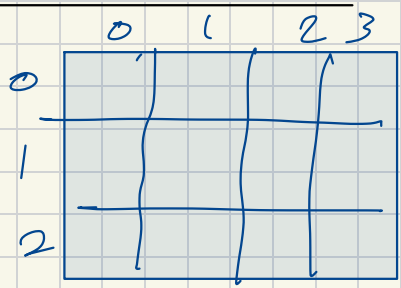
→ s







one dimension



Two Dimensional array

array[2][3]

int main ()
{
int arr[2][3] = {10, 20, 30, 40, 50, 60};

10	20	30
40	50	60

---

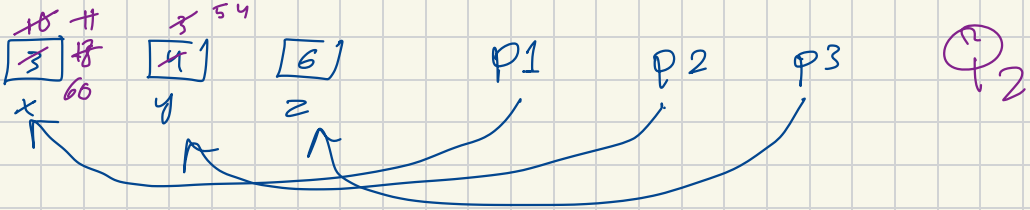
int arr[2][3] = {10, 20, 30, 40, 50, 60};

printf ("2D Array");

for (int i = 0; i < 2; i++)
{
for (int j = 0; j < 3; j++)
{
printf ("%d", array[i][j]);
}
printf ("\n");
}

i → , j ↓

10	20	30
40	50	60



num (int متغير عادي , متغير مؤشر)

```
#include <stdio.h>
# define S 5
int main()
{
    int grades[S];
    int i, sum=0, min;
    float avg;
    printf("enter five grades");
    for(i=0; i<S, i++)
        scanf("%d", &grades[i]);
    for(i=0, i<S, i++)
        sum += grades[i]
    avg = (float) sum/S,
    min = grades[0];
    for(i=0, i<S, i++)
        if (grades[i] < min)
            min = grades[i];
    printf("avg = %0.2f  min = %d", avg, min);
    return 0;
}
```

يحسب المعدل و أصغر علامة

```c
#include <stdio.h>
# define s 5
int min (int [ ] , int) ;

int main ( )
{
    int grades[s] ;
    int i , mn ,
    float avg;
    printf ("enter five grades") ;
    for (i=0; i<s ,i++)
        scanf ("%d", & grade[i]);
    mn = min (grades , s)
    printf ("min = %d" ,mn) ;
    return 0 ;
}
```

موجودة ما بنحط أقواس → &grades[0] نفس الحكي

→ ونقدر نعطي &grades[2] فيبلش بـ 2 مش 0

ما بنحط قيمة ↓

```c
int min (int g[ ] , int n)
```

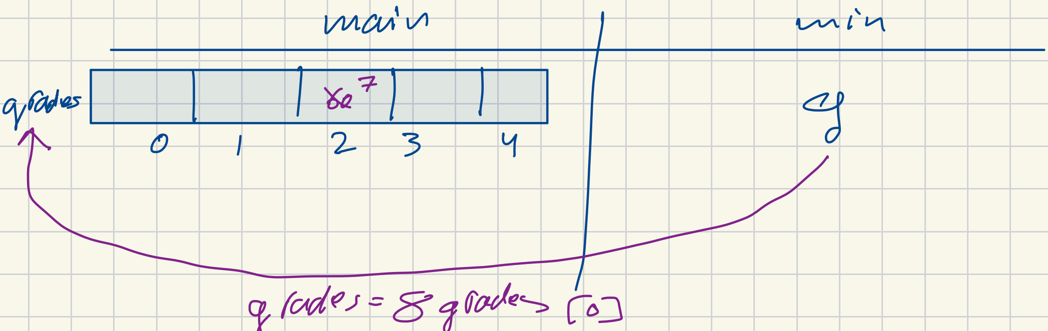
int* g معناها هيك

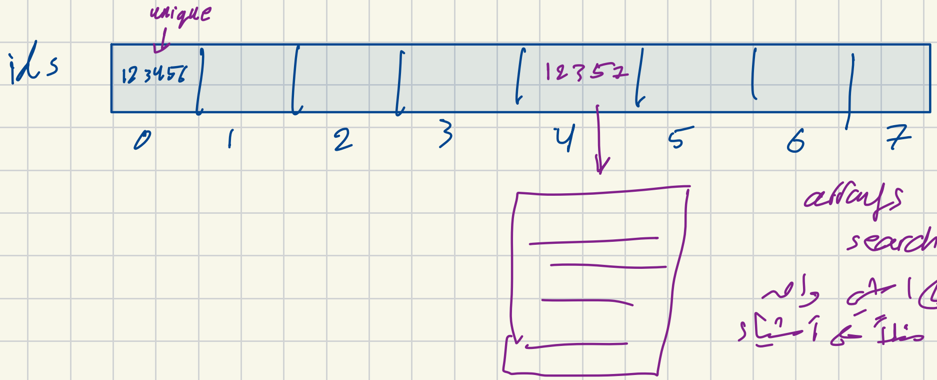
```c
{
    int m = g [0], i,
    for (i = 0 ; i<n, i++)
        if (g[i] <m)
            m = g[i] ;
    return m ;
}
```

ما بقدر أحط s → بس إذا بدنا نستخدم fn لأكثر من متغير الأفضل نستخدم n عامة ما تكون خصوصية متغير عام

→ g[2]=7;

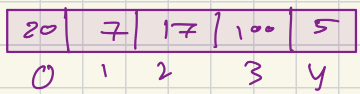
لو غيرت الـ array بالـ fn بتغير بالـ array الأصلي



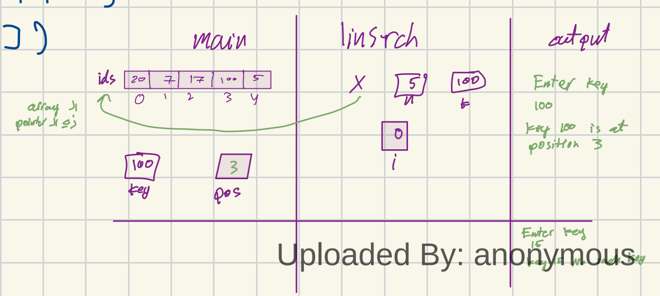


arrays
search بنستخدم
بنعدي كل الحالات واحد
بواحد مثلاً عن أشياء
أكثر

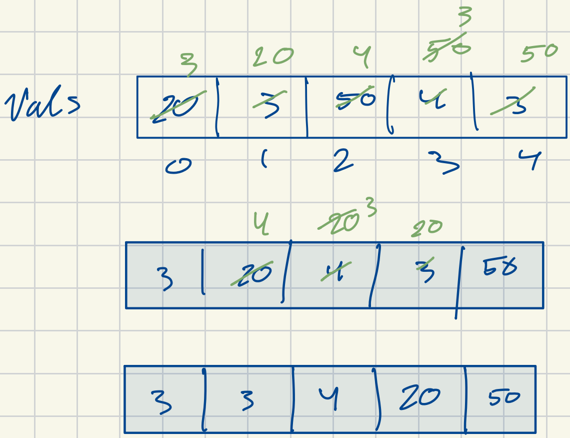
```
#include <stdio.h>
#define S 5
int linsrch (int[], int, int);
int main ()
{
    int ids [S] = { 20, 3, 17, 100, 5 };
    int key, pos;
    printf ("Enter key \n");
    scanf ( "%d", & key);
    pos = linsrch (ids, S, key);
    if (pos >= 0)
        printf (" key %d is at position %d ", key, pos);
    else
        printf (" Key %d : No such key\n", key);
    return 0;
}
int linsrch (int x[], int n, int k)
{
    int i, p;
    for (i = 0; i<n; i++)
        if (k == x [i])
            return i;
    return -1;
}
```

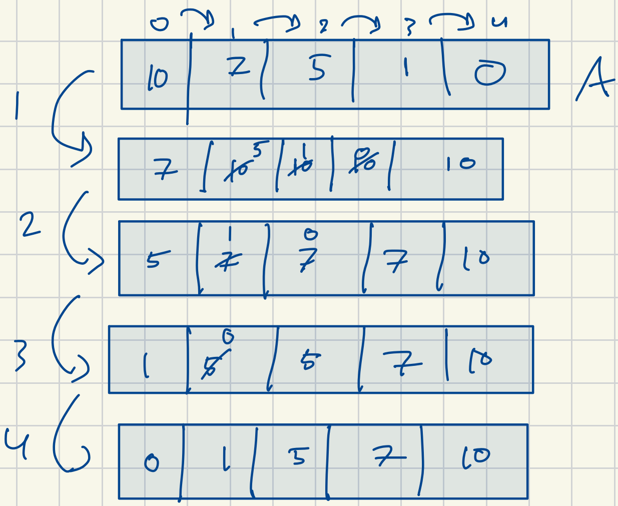


$n \rightarrow$ worst case
لو n كبير رح يسوي عمليات كثيره
ووقت كبير فبنستخدم مثلاً $\log_2 n$
أو Binary search لو ال array مرتبة



Bubble Sort





عدد المقارنات = عدد العناصر - 1

عدد column = n-1

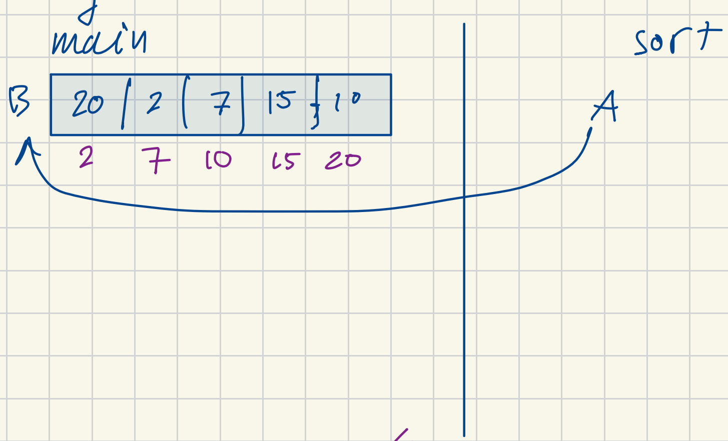
" rows = n-1

```
for (i=0; i<n-1; i++)
    for (j=0; j<n-1, j++)
        if (A[j] > A[j+1])
        {
            tmp = A[j];
            A[j] = A[j+1];
            A[j+1] = tmp;
        }
```

هاي معناه بلشنا بعنصر

```c
#include <stdio.h>
# define s 5
void sort (int [ ] , int );
int main ( )
{
    int B [s] = {20, 2, 7, 15, 10}
    int i;
    sort (B , s) ;
    for (i = 0; i < s ; i++)
        printf ("%d", B[i]);
    return 0;
}

void sort (int A[ ] , int n)
{
    for (i = 0 ; i < n-1 ; i++)
        for ( j = 0 ; j < n-1 , j++)
            if ( A [j] > A [j+1] )
            {
                tmp = A [j] ;
                A[j] = A [j+1];
                A[j+1] = tmp ;
            }
}
```

array = pointer



Sorting الوقت الوحيد إلي الـ array بكون فيه أكثر من loop

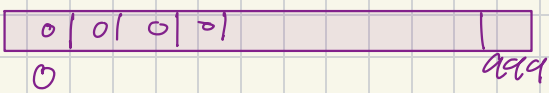
Two dimensional arrays:

← Two for loops

A	0	1	2
0	1	2	3
1	4	0	0

A [rows] [col] ⟶ A [2] [3] = {1, 2, 3, 4};

int A [1000] = {0}; ⟶ جميع العناصر تصبح أصفار



A [2] [3] = { {1, 2}, {5, 7, 1} };

A	0	1	2
0	1	2	~~0~~ 0
1	5 4	~~0~~ 7	~~0~~ 1

الأماكن الفارغة تصبح أصفار

```
#include <stdio.h>
#define R 2
#define C 3

int main ()
{ int i, j;
  int A [R] [C] = { {1,2,3}, {4,5,6} };
  int B [R] [C];
  int D [R] [C];
  printf ("Enter six values");
  for (i=0; i<R; i++)
      for (j=0; j<C; j++)
          scanf ("%d", &B [i] [j]);
```

```
for (i=0 ; i<R ; i++)
    for (j=0 , j<C , j++)
        D[i][j] = A[i][j] + B[i][j] ;
for ( i=0 ; i<R , i++)
  { for ( j=0 ; j<C ; j++)
        printf ("%d ", D[i] [j]);
    printf ("\n") ; }
return 0
}
```

ما كو بتغيرها fn تعريفها

main

A

1	2	3
4	5	6

+

B

	0	1	2
0	2	4	6
1	8	10	12

=

D

	0	1	2
0	3	6	9
1	12	15	18

out put :

```
Enter six values
2 4 6 8 10 12

3 6 9
12 15 18
```

نفس البرنامج بس ب fn.

ما كو بتعبيها

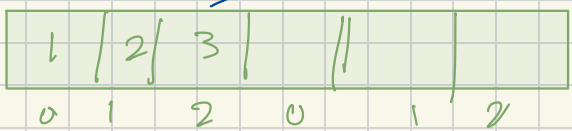
void print_Arr (int [] [C] ,int,int) : لو بنعرف ال fn فوق

```
for (i=0 ; i<R ; i++)
    for (j=0 , j<C , j++)
        D[i][j] = A[i][j] + B[i][j] ;
print_Arr (A , R , C) ;
print_Arr (B , R , C) ;
print_Arr (D , R , C) ;
printf ("\n + \n") ;
  "  ("\n = \n") ;
```



```
void print_Arr (int x[][c], int r, int c)
{
    int i, j;
    for (i=0; i<r; i++)
    {  for (j=0; j<c; j++)
           printf ("%d ", x[i][j]);
       printf ("\n");
    }
}
```

small / Capital

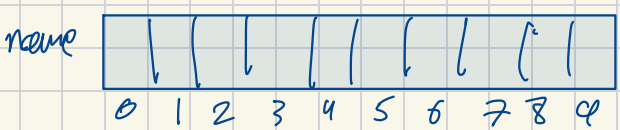


---

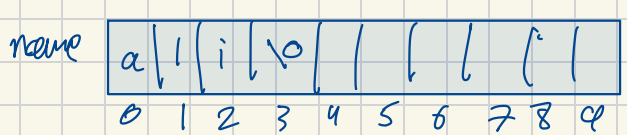
strings :

int
float
char

char name [10]

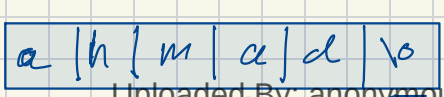


char name [10] = "ali";



\0 ← null char

char name [] = "ahmad";



data.txt

60
90
50
40
⋮

ما بنعرف عدد القيم

```
FILE *in;
float avg;
int count=0, grade, status, sum=0;
in = fopen("grades.txt", "r");
status = fscanf(in, "%d", &grade);
While (status != EOF)        → End of File
{
    sum += grade;
    count ++;
    status = fscanf(in, "%d", &grade);
}
avg = (float) sum/count;
printf("avg = %0.0f", avg);
fclose(in);
return 0;
```

```
char name[10];
printf("Enter your nam");
scanf("%s", name);   ⟹
printf("Hi %s", name);
```

%s → هاي لل string

name → اسم هو arr address فش داعي &

لو salma_Dalbah يكتب Hi Salma

بنبدل هاي ب gets(name); عشان نكتب اسم كامل

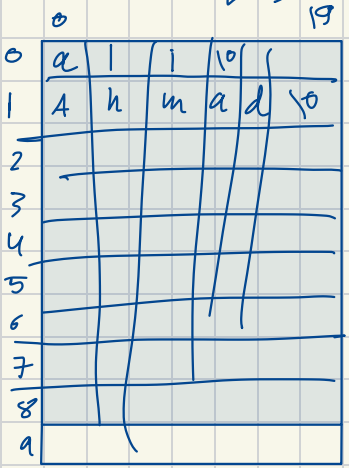
```
char name[10] = "ahmad";
count = 0;
i = 0;
While (name[i] != '\0')
{
    if (name[i] == 'a')
        count ++;
    i++;
}
```

a	h	m	a	d	\0				

0 … 9

```
#include <stdio.h>
#define S 10
int countA(char []);
int main()
{
  char name[10] = "alimcal";
  int count;
  count = countA(name);
  printf("number of a's in name = %d", count);
  return 0;
}

int countA(char x[])
{
   int i, cnt = 0;
   i = 0;
   while (x[i] != '\0')
       if (x[i] == 'a')
           cnt++;
   return cnt;
}
```

```
char name[10][20] = {"ali", "ahmad", ...};
```



```
if (name == "ahmad")
name1 = name2
```

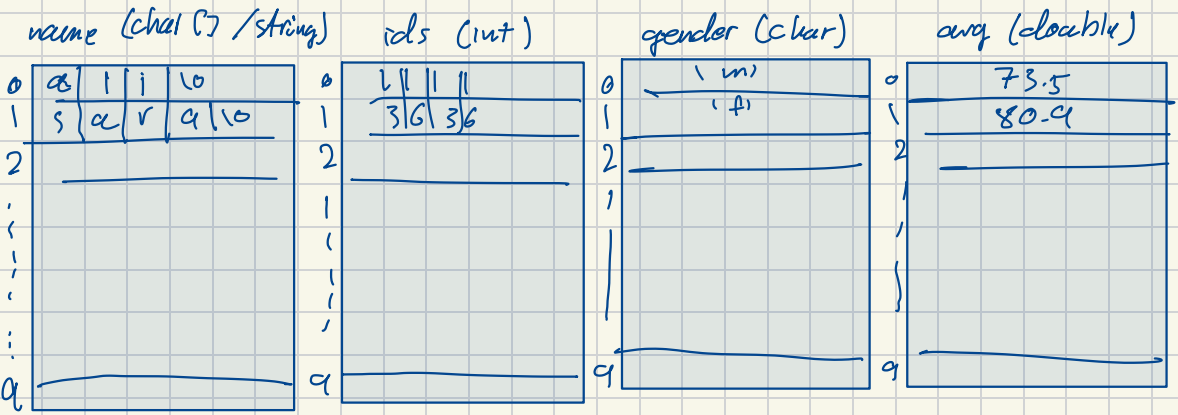
```
strcmp (name, "ahmad");
```

<0 =0 >0

```
strcpy (name, neame);
```

```
char name [10] [20] = {"ali", "ahmad", ...};
for (i=0; i<10, i++)
{
  printf ("Enter name");
  scanf ("%os", name[i]);
}
```

Parallel Arrays




```
char name [10] [20];
int ids [10];
char gends [10];
double avg [10];
```