Arrays

int A[5];

A					
index	0	1	2	3	4

↙ index (subscript)

int A[5] = {3, 2, 7, 5, 4};

A					
	3	2	7	5	4
	0	1	2	3	4

int A[5] = {1, 5};

1	5	0	0	0
0	1	2	3	4

int A[ ] = {5, 7};   A[2] = {1, 2, 7}; → This is wrong

A		
	5	7
	0	1

int A[5] = {0}; ← حالة خاصة

• **By user** :-

```
int grades [4];
int i;
for (i=0 ; i<4 ; i++)
{
    printf ("Enter grade\n");
    scanf ("%d", & grades[i]);
}
```

grades	7̶0̶ 69	1̶0̶0̶ 9̶8̶ 7	60	9
	0	1	2	3

X=3;

operations

grades[1] = grades[1] - 2

grades[0] = grades[2] + grades[3]

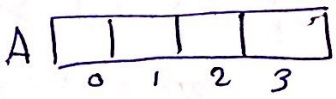
grades [X-2*1] = 7;

Enter grade
70
100
60
9

Lecture 22

Basic arrays

```
#include <stdio.h>
#define S 4

int main ( )
{
   int A [S] = { 5, 2, 7 };
   int sum = 0, i, max, avg;
   for (i=0; i<S; i++)
      sum += A[i]
      avg float sum/S;
   max = A[0];
   for (i=0; i<S; i++)
      if (A[i] > max)
      max = A[i];
      {
}
```



Arrays with functions :-

```
#define S 4
int max( int[ ], int);
int main
{
int A[S] = {5, 6, 2};
int m;
```

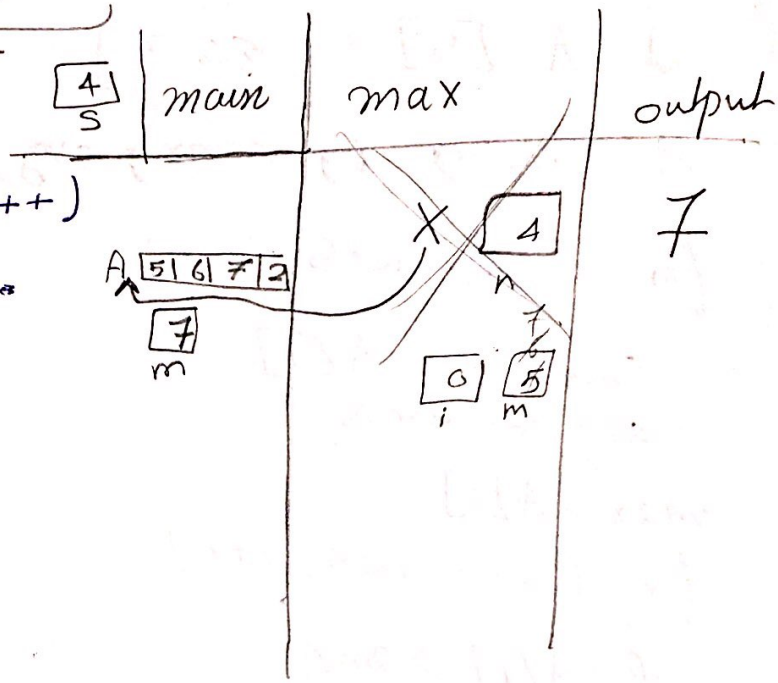
```
m = max (A, S);
Printf ("max = %d", m);
return 0;
}
```

size

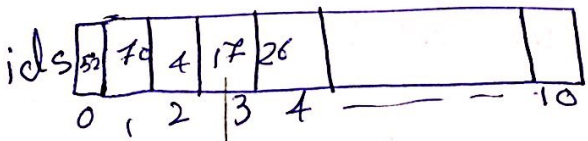
A = &A[0]

اسم الدالة هو العنوان تاع أول عنصر فيها

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



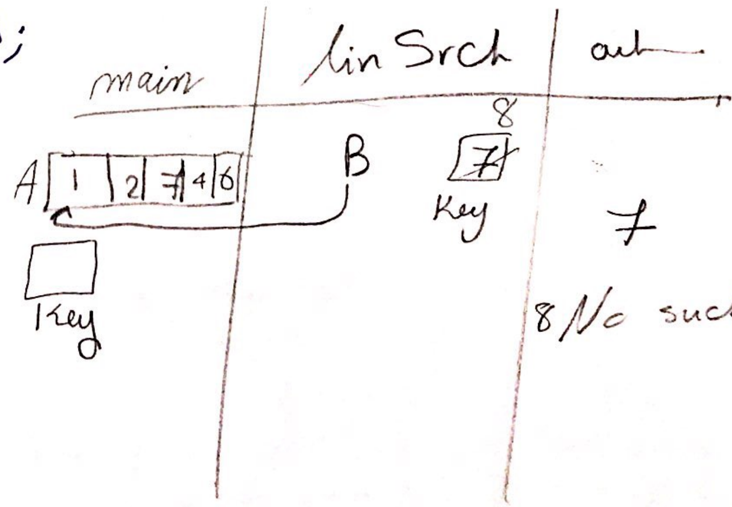
linear search



- if the ids is 17

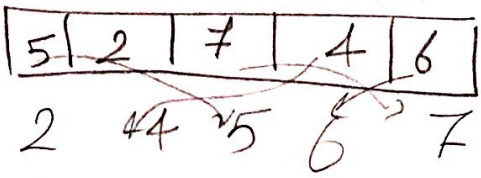
search for Binary search

```
clude <Stdio.h>
define S 5
int linSrch (int [ ], int, int);
int main ( )

{
  int Key, A[S] = {1,2,7,4,6};
  Printf (" Enter Key\n")
  scanf (" %d ", &Key);
  pos = linSrch (A, S, Key);
  if (pos == -1)
      Printf ( %d No such Key", Key);
  else Printf (" %d is at position %d, Key, pos);
}

int linSrch (int B[ ], int n, int K)
{
  int i;
  for (i = 0; i < n; i++)
  if (B[i] == K)
      return i;
  return -1;

}
```



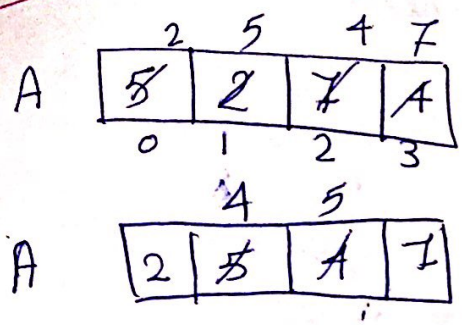
→ Sorting

↳ Bubble sort

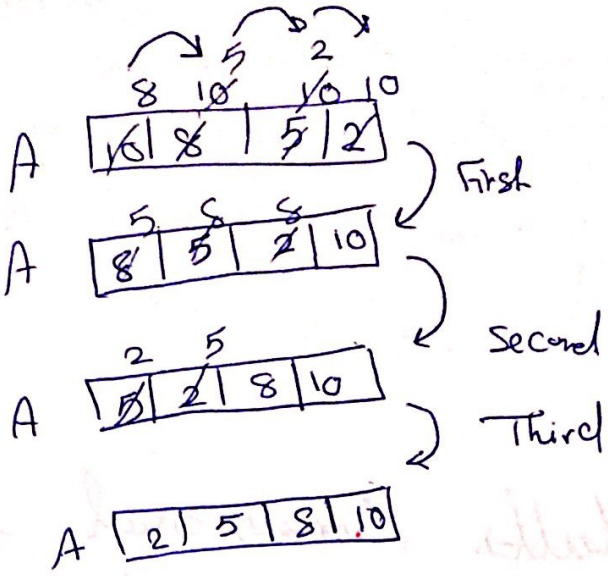


2 4 5 6 7

Bubble sort



Worst Case



- We need two loops

```
(function) Void Sort (int A[ ], int n) → int i, j, temp;
    for (i=0; i<n-1; i++)
        for (j=0; j<n-1; j++)
            if (A[j] > A[j+1])
            {
                temp = A[j];
                A[j] = A[j+1];
                A[j+1] = temp;
            }
```

- Arrays with characters :-

```
int Ecount = 0;
Char letter [6] = {'a', 'b', 'x', ---};
PrintF ("Enter word")
for (i=0; i<n; i++)
    scanf ("%c", &letters[i]);
```

→

letters

h	e	l	l	o	
0	1	2	3	4	5

Ex:
Enter word
hello

```
for (i=0; i<n; i++)
    if (letter[i]=='e')
    Ecount++;
```

• لحساب عدد حروف e في الكلمة

• Multi dimensional Arrays:-

→ 2 dimensional Arrays :-

```
int A[2][3];
```

rows, columns

A	0	1	2
0	5	2	0
1	3	4	6

```
int A[2][3] = { {5,2,0}, {3,4,6} };
```

2×3 (2 Rows, 3 Cols)

• To enter elements by user :-

```
#include <stdio.h>
#define R 2
#define C 3
int main()
{
    int A[R][C];
    int i, j;
    for(i=0; i<R; i++)
      for (j=0; j<C; j++)
```

```
printf ("Enter value\n");
scanf ("%d", & A[i][j]);
}
```

Enter value
5 A[0][0]
Enter value
10 A[0][1]

E______
3 A[0][2]

E______
6 A[1][0]

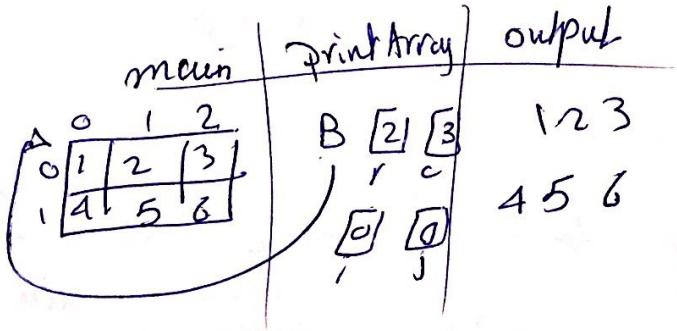
E______
2 A[1][1]

E______
1 A[1][2]

A	0	1	2
0	5	10	3
1	6	2	1

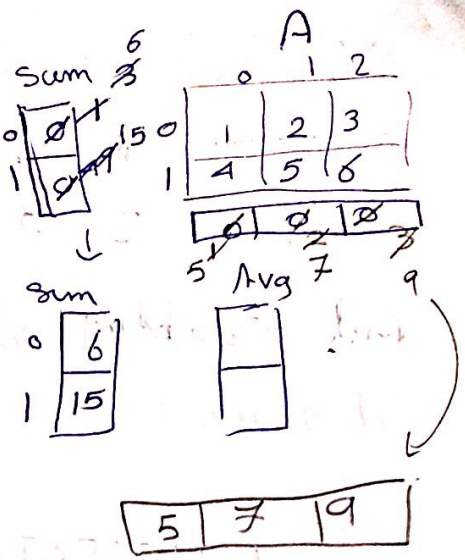
```
for (i=0; i<R; i++)          → To sum
  for (j=0; j<c; j++)
  {
     sum += A[i][j];

  max = A[0][0];              → To Choose Max
  for (i ——— same )
    for (j ——— same )
    { if ( A[i][j] > max)  →  max A[i][j];
```

```
#include <stdio.h>
#define R 2
#define C 3
void printArry(int [ ][C], int, int);
int main ( )
{
  int A[R][C] = { {1,2,3}, {4,5,6} };
   printArray (A, R, C);
   return 0;
}

Void printArray (int B[ ][C], int r, int c)
{
  int i, j;
  for (i=0; i<r; i++)
  {
    for (j=0; j<c; j++)
    printf ("%d", A[i][j]);
    printf ("\n");
  }
}
```



- A code to Sum every Column

```
int sum[R] = {0};
for(i=0; i<R; i++)
  for(j=0; j<C; j++)
    Sum[i] += A[i][j];
```



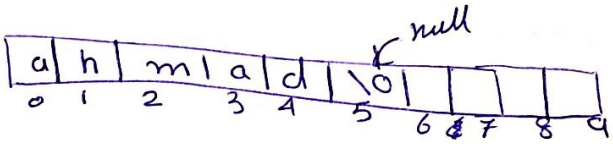
To find Avg :-

Use another loop with

```
Avg[i] = Sum[i]/3.0;
```

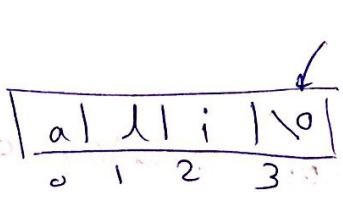
• strings

- it's a character Array
- Char name[10] = "Ahmad"



size > characheres + null

Char name[ ] = "ali";

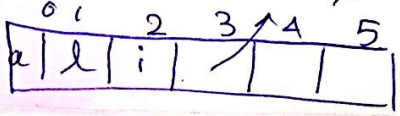


'u' u

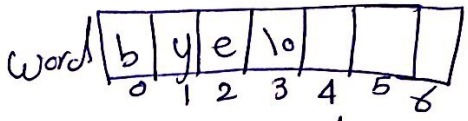
"u" u\0

Char name[6] = {'a', 'l', 'i'};

No null



```
Char Word[7];
Printf ("Enter Word\n");
scanf ("%s", word);        Word = &word[0]
          ↑ no adress
Printf ("Word is %s", Word);
```



Enter Word
bye
Word is bye

char $S_1[]$, $S_2[]$

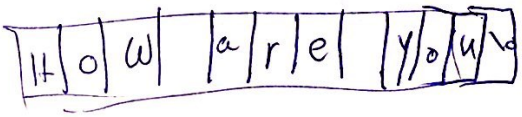
you can't do :- $S_1 = S_2$; ✗

→ if ($S_1 == S_2$) ✗

Enter Word
ali ali
space
in this case don't use scanf
use function (gets)

gets (word)

Ex: Char sent[100];
gets (sent)



String functions

#include <string.h>

strcpy (S1, S2);

strncpy (S1, S2, n);

strcat (S1, S2);

strncat (S1, S2, n);

strcmp (S1, S2);

strncmp (S1, S2, n);

strlen (S1);

strtok (_ , _);

Note:

ممنوع نضع constant هنا — لا يكون الا variable

strcpy (S1, "sami")

"sami" → constant

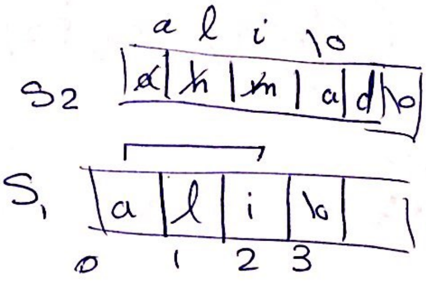
strcpy للتبديل

char S1[10] = "ali";

char S2[6] = "ahmad";

S2 = S1; ✗

strcpy (S2, S1);

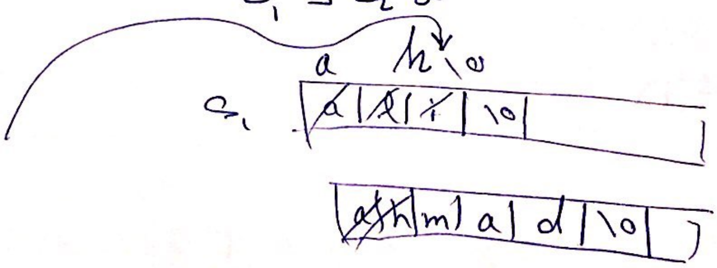


لتبديل عدد معين من elements

strncpy (S1, S2, 2); ينسخ اول 2 char من S2 لـ S1

لا تضع null

S1[2] = '\0';



strcat (s1, s2) 2 Arrays

strcat (s1, s2);

ali ah \0

s1 | a | l | i | \0 a | h | \0 | |

strncat

strncat (s1, s2, 1);

s1 | a | l | i | \0 a | \0 | |

strcmp

ترجع integer — "ahmad"

s1 ahmad

s2 ahlam

if (strcmp (s1, s2) == 0 / > 0 / < 0)

يجب دمج التكملة لكي نستطيع مقارنة الشي الذي يرجع من ال function

- < 0 : s1 < s2
- == 0 : s1 = s2
- > 0 : s1 > s2

strncmp (s1, s2, 2) == 0

مقارنة جزء

```
s = strlen ("ahmaad")
```
لا يحسب الـ null

```
printf ("%d", s)
```
فهنا ترجع خمسة

→ I got here

• اذا أردت ان أكتب عدد حروف الـ string

```
int i=0; countA=0;
char st[10];
printf ("Enter string\n");
scanf ("%s", st);

while (st[i] != '\0')
{
   if (st[i]=='a')
       countA++;

   i++;
}
```

a	h	a	b	\0						
0	1	2	3	4	5	6	7	8	9	

```
i = strlen(st);
```
→ طريقة أسرع
تشيل
مع الـ null

التطويل

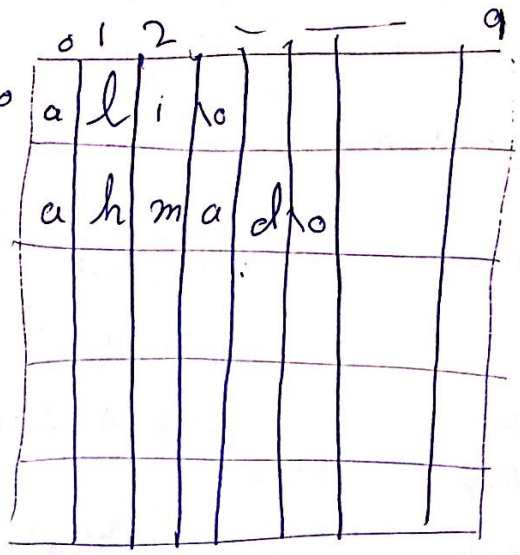
```
while (st[i] != '\0')
    i++;
```

Char names[5][10];

:

```
for (i=0; i<5; i++)
  {
  printf ("Enter name\n");
  scanf ("%s", names[i]);
  }
```

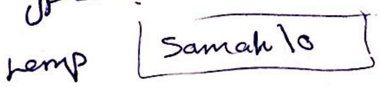
names[i]



```
#include <stdio.h>
#include <string.h>
#define S 5

int main ( )
{
 Char name [S] [10], temp[10];
 int i , j,
 Printf (" Enter any five names\n");
  for (i=0; i<s ; i++)
     Scanf ("%S", names [i]);

  for (i=0; i<S-1 ; i++)
     for (j=0; j<S-1; j++)

  ~~If (Vates [j] > Vates [j+1])~~
  If ( strcmp (names[j], names[j+1]) >0)

     ~~String (~~
     strcpy (temp, names [j]);
      strcp( names[j], names [j+1]);
      strcp ( names [j+1], temp );
```

one dimension Arry ذي هاي الأراي

Bubble Sort الـ طريقة



تتغير

Strtok : (____ , ____);

↗ tokenisor

ex : how, are, you   يأخذ منها قطع


```
#include <stdio.h>
#include <string.h>
#define S 100

int main ( )
{
    Char sent [S];
    Char * token ;
    printf (" Enter sentence \n");
    gets (sent);
    token = ~~string~~ strtok (sent, " a ");
```

زي ال

مثلا اذا لقيت space أو نقطة أو ___ اقطع

```
    while (token != NULL)
    { printf (" %s\n" , token);
      token = strtok (NULL, " _ ");
    }

    return 0;
}
```

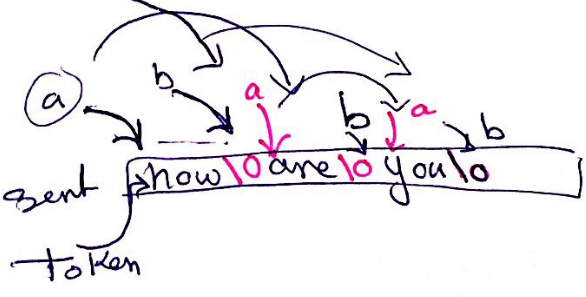
(token != NULL) → يعني ما في اشي

printf → يطبع → how / are / you

" _ " → space

token = strtok (NULL, ...) → ??

هون الاختلاف



Important

ex
Input :-
How are you

```
#include < stdio.h>
# include < string.h>
# define S 100
int main ()
{
  char sent[s], words[10][20];
  char *token;
  printf ("Enter sentence \n");
  gets( sent);
  token = strtok (sent, " ").
  while (token != NULL).
  {
     strcpy ( words [i++] , token);
       token = strtok( NULL, " ").
  }
  return 0;
}
for (j = i-1; j>=0; j--)
printf ("%s ", words [j]
```

```
if(!(strcmp (token, "is") ==0)
if(!(strcpy (word[i], "are")).
strcpy(word[i], "is").
```

i ③

	Words
0	how \0
1	~~are \0~~ is \0
2	you \0
3	

نطبع ←

you are how