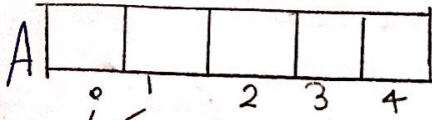


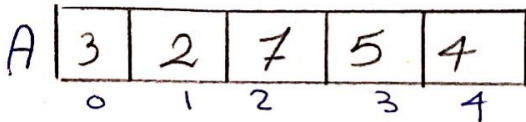
Arrays

```
int A[5];
```

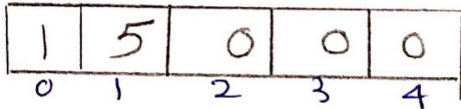


index (subscript)

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



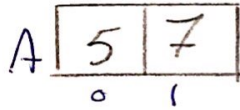
```
int A[5] = {1, 5};
```



```
int A[] = {5, 7};
```

$A[2] = \{1, 2, 7\}$

This is wrong



```
int A[5] = {0};
```

By user :-

```
int grades[4];
```

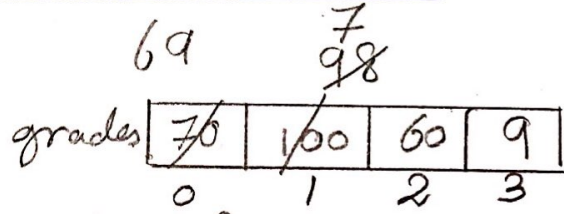
```
int i;
```

```
for (i=0; i<4; i++)
```

```
{
```

```
printf("Enter grade\n");
scanf("%d", &grades[i]);
```

```
}
```



operations
 $grades[1] = grades[1]$
 $grades[0] = grades[2] - 2 + grades[3]$
 $grades[x-2*1] = 7;$

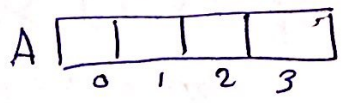
Enter grade
 70
 ~
 100
 ~
 60
 ~
 9

ture 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, 7, 2 };
  int m;
```

```

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

```

A = &A[0]
 اسم الذاكرة هو العنوان
 تابع اول عنصر فيها

```

int max(int X[], int n)

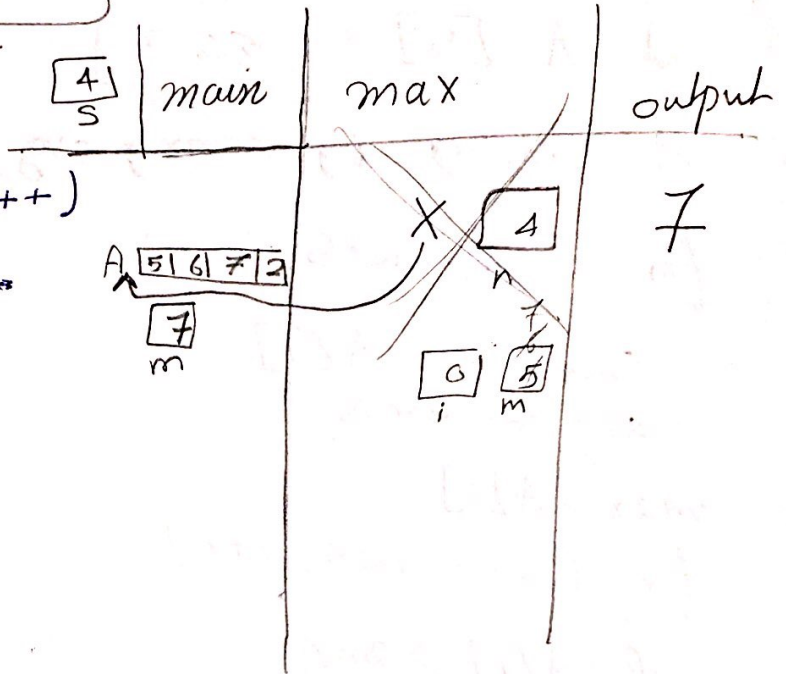
```

Array غير مرتب
 تنظير
 في

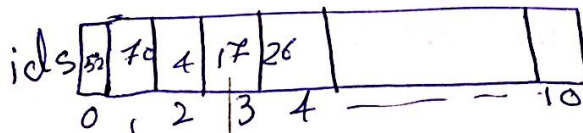
```

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
 ↓
 a Record



search for
Binary search

```

#include <stdio.h>
#define S 5
int linSrch (int [ ], int, int);
int main ( )

```

```

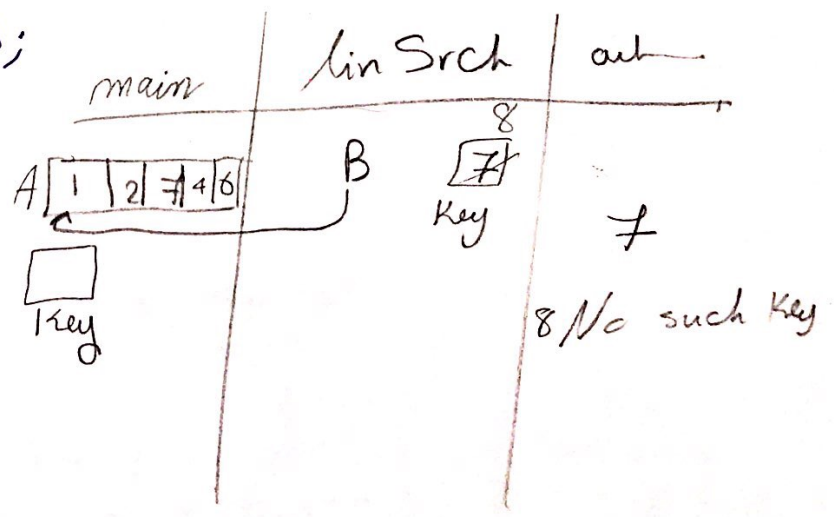
{
    int key, A[S] = { 1, 2, 7, 4, 6 };
    printf (" Enter key\n");
    scanf ("%d", &key);
    pos = linSrch (A, S, key);
    if (pos == -1)
        printf ("%d No such key", key);
    else
        printf ("%d is at position %d", key, pos);
}

```

```

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

```



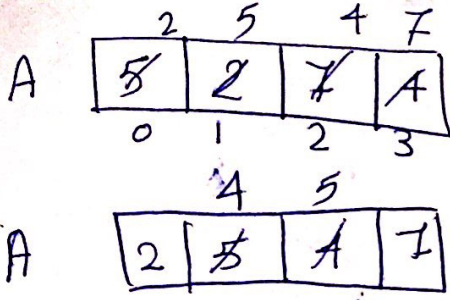
→ Sorting

↳ Bubble sort

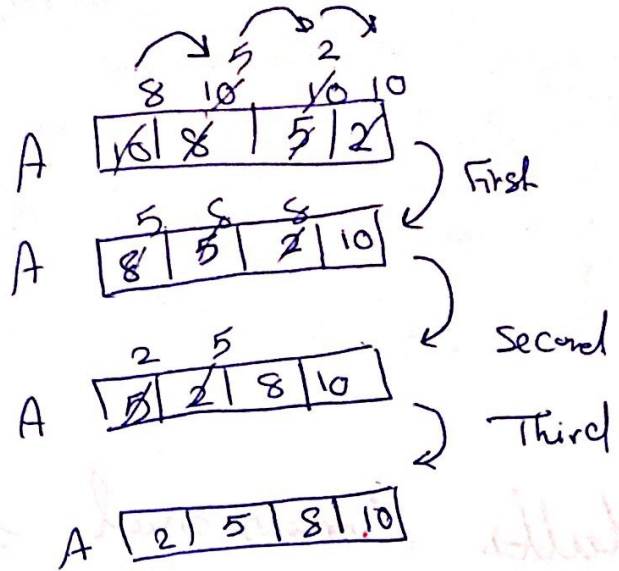
5	2	7	4	6
---	---	---	---	---

2 4 5 6 7

Bubble sort



Worst case



• We need two loops

```

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

• Arrays with characters :-

```

int count = 0;
char letter [6] = {'a', 'b', 'x', '---'};

printf ("Enter word")
for (i=0; i < n; i++)
    scanf ("%c", &letter[i]);
    
```

letters

h	e	l	l	o
0	1	2	3	4

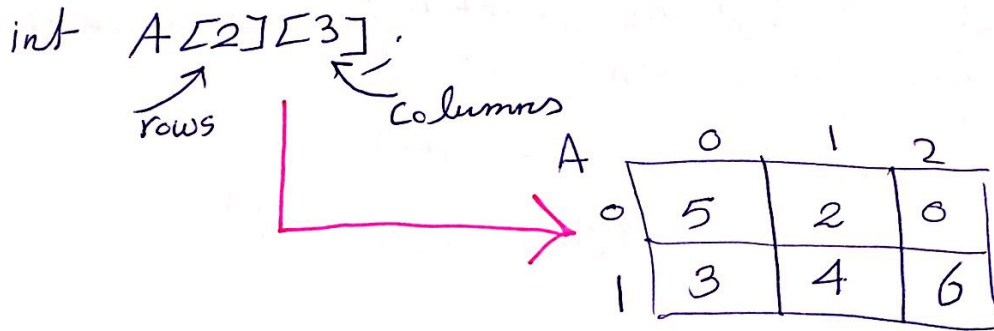
Ex:
UPLOADED BY AHMAD
Enter word
hello

```

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

لعداد عدد حرف e في الـ string.

• Multi dimensional Arrays:-
→ 2 dimensional Arrays:-



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

2x3
2 Rows 3 Col

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]

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

E
 3 A[0][2]

E
 6 A[1][0]

E
 2 A[1][1]

E
 1 A[1][2]

```

for (i=0; i<R; i++)
  for (j=0; j<C; j++)

```

→ sum

```

{
  sum += A[i][j];
}

```

```

max = A[0][0];

```

→ 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 printArray (int [ ][C], int, int);

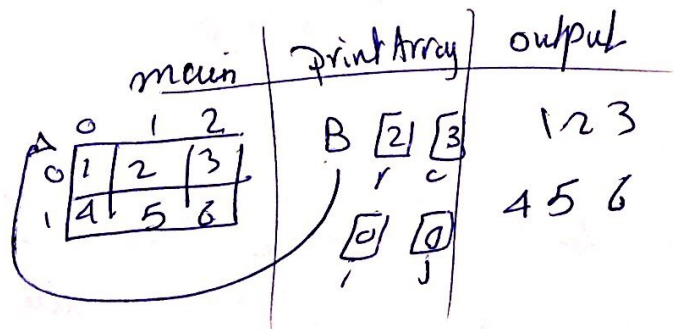
int main ( )
{
    int A[R][C] = { {1, 2, 3}, {4, 5, 6} };
    printArray (A, R, C);
    return 0;
}

```

```

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\t", A[i][j]);
        printf ("\n");
    }
}

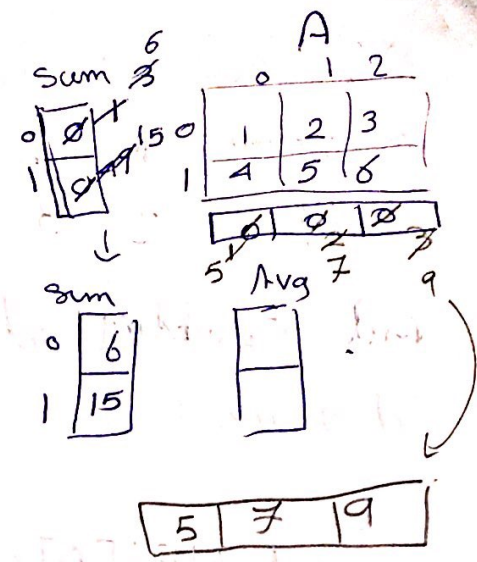
```



• A Code to Sum every column UPLOADED BY AHMAD JUNDU

```
int sum [R] = {0};
```

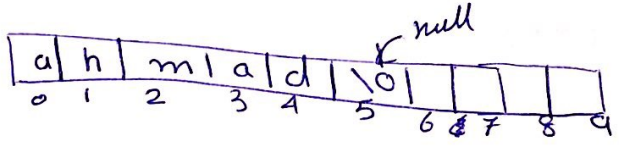
```
for (i=0; i < R; i++)
    for (j=0; j < C; j++)
        Sum [i] += A [i] [j];
```



→ To find Avg: Use another loop with
 $Avg [i] = Sum [i] / 3.0;$

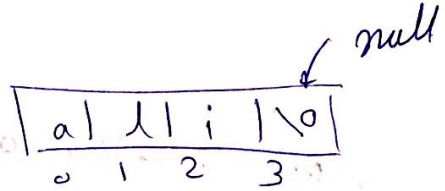
strings

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



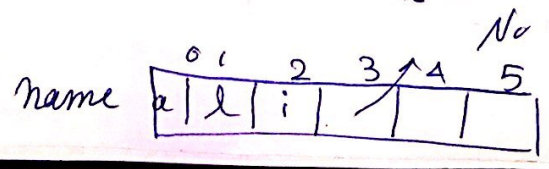
size > characters + null

```
Char name [ ] = "ali";
```



u
 ~ u ~ (u\0)

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



Char Word [7];

Printf ("Enter Word\n");

scanf ("%s", word);
no address

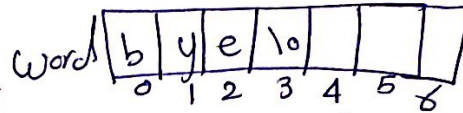
word = s word [0]

Printf ("Word is %s", word);

char s₁[], s₂ []

you can't do ~~s₁ = s₂~~

→ if (s₁ == s₂) ~~X~~



Enter word
bye

word is bye

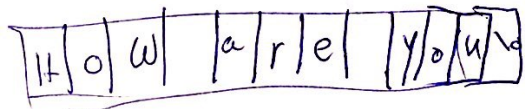
Enter word

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>

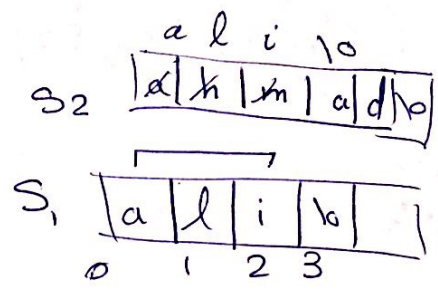
- strcpy (s1, s2);
- strncpy (s1, s2, n);
- strcat (s1, s2);
- strncat (s1, s2, n);
- strcmp (s1, s2);
- strncmp (s1, s2, n);
- strlen (s1);
- strtok (-, -);

Note :

constant vs variable
 strcpy (s1, "samir")
 ↑
 constant

strcpy

```
char s1[10] = "ali";
char s2[6] = "ahmad";
s2 = s1; // ✗
strcpy (s2, s1);
```

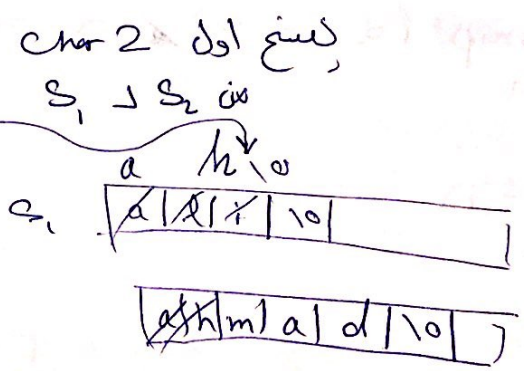


تبدیل (conversion) elements

strncpy (s1, s2, 2);

↑
 null ختم

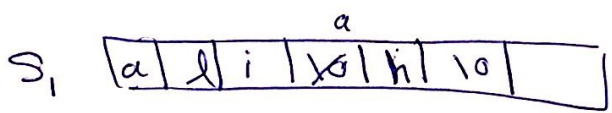
```
s1[2] = '\0';
```



strcat (s1, s2)

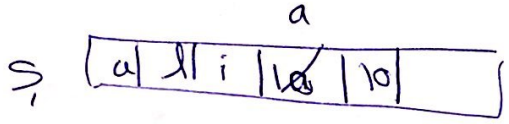
```
strcat (s1, s2);
```

alich 10

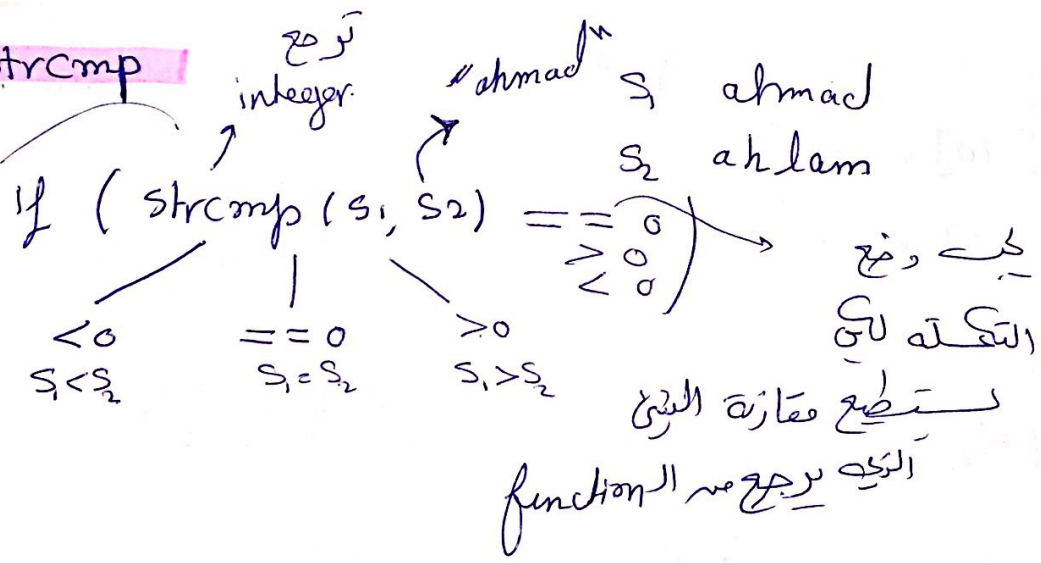


strncat

```
strncat (s1, s2, 1);
```



strcmp



```
strcmp (s1, s2, 2) == 0
```

↑
مقارنة جزئية

```

len ("ahmaach")
نصف حساب
printf ("%d", s)
نصف ربع خمسة

```

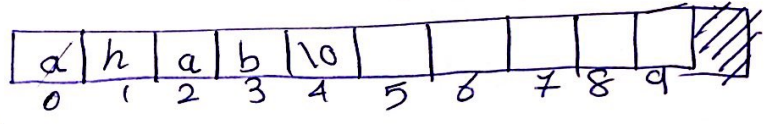
I got here

• اذا أردت ان تحسب عدد حروف

```

string s;
int i=0; count A=0;
char st[10];
printf ("Enter string\n");
scanf ("%s", st);
while (st[i] != '\0')
{
    if (st[i] == 'a')
        count A++;
    i++;
}

```



```

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

```

صديقك اصنع
تبريلي
صفحة الـ null

تسوية

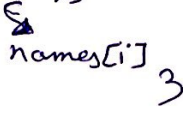
Char names [5][10];

```
for (i=0; i<5; i++)
```

```
{ printf ("Enter name\n");
```

```
scanf ("%s", names[i]);
```

```
}
```



	0	1	2					9
0	a	l	i	n				
1	a	h	m	a	d			
2								
3								
4								

25:-

```

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

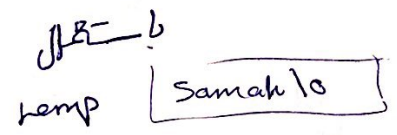
int main ()
{
  char name [S][10], temp[10];
  int i, j;
  printf ("Enter any five names\n");
  for (i=0; i<S; i++)
    scanf ("%s", names [i]);

  for (i=0; i<S-1; i++)
    for (j=0; j<S-1; j++)
      if (strcmp (names [j], names [j+1]) > 0)

```

0	s	a	m	a	h	l	o
1	A	l	i	l	o		
2	Z	i	n	g	l	e	
3	e	m	a	n	l	o	
4	a	a	a	a	a	a	

one dimension Array [s][10] ←



```

strcpy (temp, names [j]);
strcpy (names [j], names [j+1]);
strcpy (names [j+1], temp);

```



strtok: (— , —):

↑
tokenizer

ex: how, are, you

كيفية

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

```
int main ()
{
```

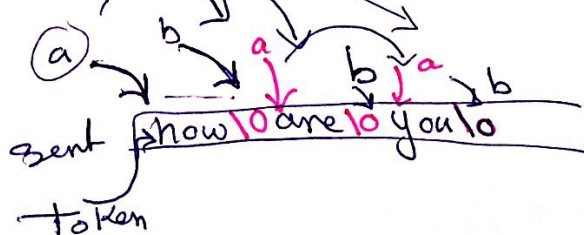
```
    char sent[S];
    char * token;
    printf("Enter sentence\n");
    gets(sent);
    token = strtok(sent, " ");
```

قائمة الأجزاء أو فواصل أو قطع

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

how
are
you

هنا الاختلاف



```

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

```

Important

ex
Input :-
How are you

```

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, " ");
    }
}

```

Words	
0	how to
1	are to
2	you to
3	

```

return 0;
}

for (j = (i-1); j >= 0; j--)
    printf ("%s \n", words[j])

```

~~strcpy (word[i], "are");~~
 strcpy (word[i], "is");

•
you are how