

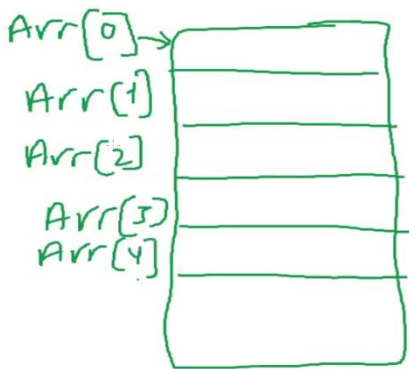
Array

- هي عبارة عن Derived data type
- وهي بتبقى عبارة عن شيء بحفظ فيه مجموعه variables بس بشرط (لازم) يكونوا نفس النوع
- بتتمثل في الميموري ورا بعض وعلشان الميزه دي ممكن تعمل access لاي عنصر فيه وببيدء من الصفر

Syntax:

Data type name [size];

Int arr [5];



الاقواس المربعة بنسميها Subscriptor

الArray :

1. ال index بتاعها ببيدء من الصفر
2. ال index بيتكتب بين ال []
3. اخر index بيساوي ال size -1 لانه ببيدء من الصفر
4. ال length لازم يكون constant يعني مينفعش متغير (Variable length array متاح في ال C99)

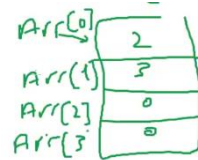
Initialization:

1. int arr [5] = {1,2,3,4,5};
2. int arr [5];
int arr[0]=1;
int arr[1]=2;
int arr[2]=3;
int arr[3]=4;
int arr[4]=5;

لو حطينا ال size برقم ما مثلا 4 ثم في ال initialization اديتله {1,2} بس :
الي هيجصل انه هيكمل الباقي باصفار

```
1 #include <stdio.h>
2 //include "functions.h"
3 int main(void)
4 {
5     int arr[6]={0};
6     for(int i=0;i<6;i++)
7     {
8         printf("the element num %d = %d\n",i,arr[i]);
9     }
10    return 0;
11 }
```

```
C:\Windows\System32\cmd.exe
no: gcc "project 1.c"
no: a.exe
the element num 0 = 0
the element num 1 = 0
the element num 2 = 0
the element num 3 = 0
the element num 4 = 0
the element num 5 = 0
no: .
```



فممكن اعرف Array كلها اصفار بالطريقة دي

طيب لو زدنا عن ال size

مثال:

```
int arr[5]={1,2,3,4,5,6};
```

ده معناه انه ينفذ الكود بتاعك عادي ممكن يدبلك تحذير لكن هينفذه وبالتالي انت ممكن تغير في قيمه متغير من غير ما تاخذ بالك

وال length هيفضل ثابت

ودي امثله لكود شغال ب Array

```
1 #include <stdio.h>
2 //#include "functions.h"
3 int main(void)
4 {
5     int arr[10];
6     for(int i=0;i<10;i++)
7     {
8         printf("Enter element num %d :",i+1);
9         scanf("%d",&arr[i]);
10    }
11    int sum=0;
12    for(int i=0;i<10;i++)
13    {
14        sum+=arr[i];
15    }
16    int avr=sum/10;
17    printf("the sum is %d\n",sum);
18    printf("the avr is %d",avr);
19    return 0;
20 }
```

```
C:\Windows\System32\cmd.exe
mo: gcc "project 1.c"
mo: a.exe
Enter element num 1 :10
Enter element num 2 :20
Enter element num 3 :30
Enter element num 4 :40
Enter element num 5 :50
Enter element num 6 :60
Enter element num 7 :70
Enter element num 8 :80
Enter element num 9 :90
Enter element num 10 :100
the sum is 550
the avr is 55
mo:
```

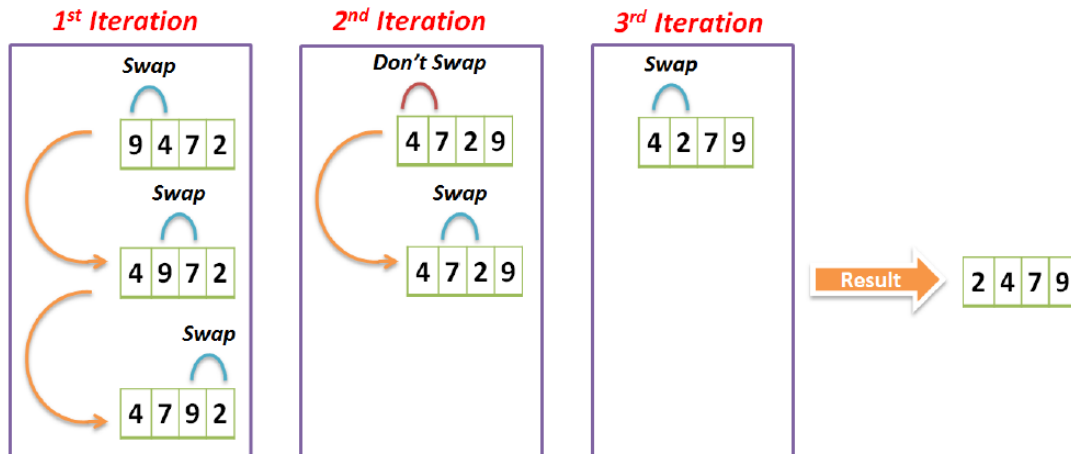
```
1 #include <stdio.h>
2 //#include "functions.h"
3 int main(void)
4 {
5     int arr[10];
6     for(int i=0;i<10;i++)
7     {
8         printf("Enter element num %d :",i);
9         scanf("%d",&arr[i]);
10    }
11    for(int i=9;i>=0;i--)
12    {
13        printf("the num%d is : %d\n",i,arr[i]);
14    }
15    return 0;
16 }
17 }
```

```
C:\Windows\System32\cmd.exe
mo:
mo: a.exe
Enter element num 0 :1
Enter element num 1 :2
Enter element num 2 :3
Enter element num 3 :4
Enter element num 4 :5
Enter element num 5 :6
Enter element num 6 :7
Enter element num 7 :8
Enter element num 8 :9
Enter element num 9 :1
the num9 is : 1
the num8 is : 9
the num7 is : 8
the num6 is : 7
the num5 is : 6
the num4 is : 5
the num3 is : 4
the num2 is : 3
the num1 is : 2
the num0 is : 1
mo: _
```

هنراجع ال bubble sorting algorithm علشان نطبق اكتر:

بيعمل ايه هو ببساطه انت بتمشي في ال Array لو العنصر الي انا واقف عليه اقل من العنصر الي قبله بنبدلهم لكل ال Array ودي بنسميها اول لفه وبعد كده بنكررها ثاني طيب بنكررها كام مره ← عدد عناصر ال Array 1-

Bubble Sorting Concept



Notes:

Number of iterations = Size of array – 1

التطبيق:

```

1  #include <stdio.h>
2  // #include "functions.h"
3  int main(void)
4  {
5      int temp;
6      int arr[10];
7      for(int i=0; i<10; i++)
8      {
9          printf("Enter element num %d :", i+1);
10         scanf("%d", &arr[i]);
11     }
12
13     for(int i=0; i<10; i++)
14     {
15         for(int y=0; y<9; y++)
16         {
17             if(arr[y]>arr[y+1])
18             {
19                 temp=arr[y];
20                 arr[y]=arr[y+1];
21                 arr[y+1]=temp;
22             }
23         }
24     }
25     printf("the array is : ");
26     for(int i=0; i<10; i++)
27     {
28         printf("%d ,", arr[i]);
29     }
30     return 0;
31 }

```

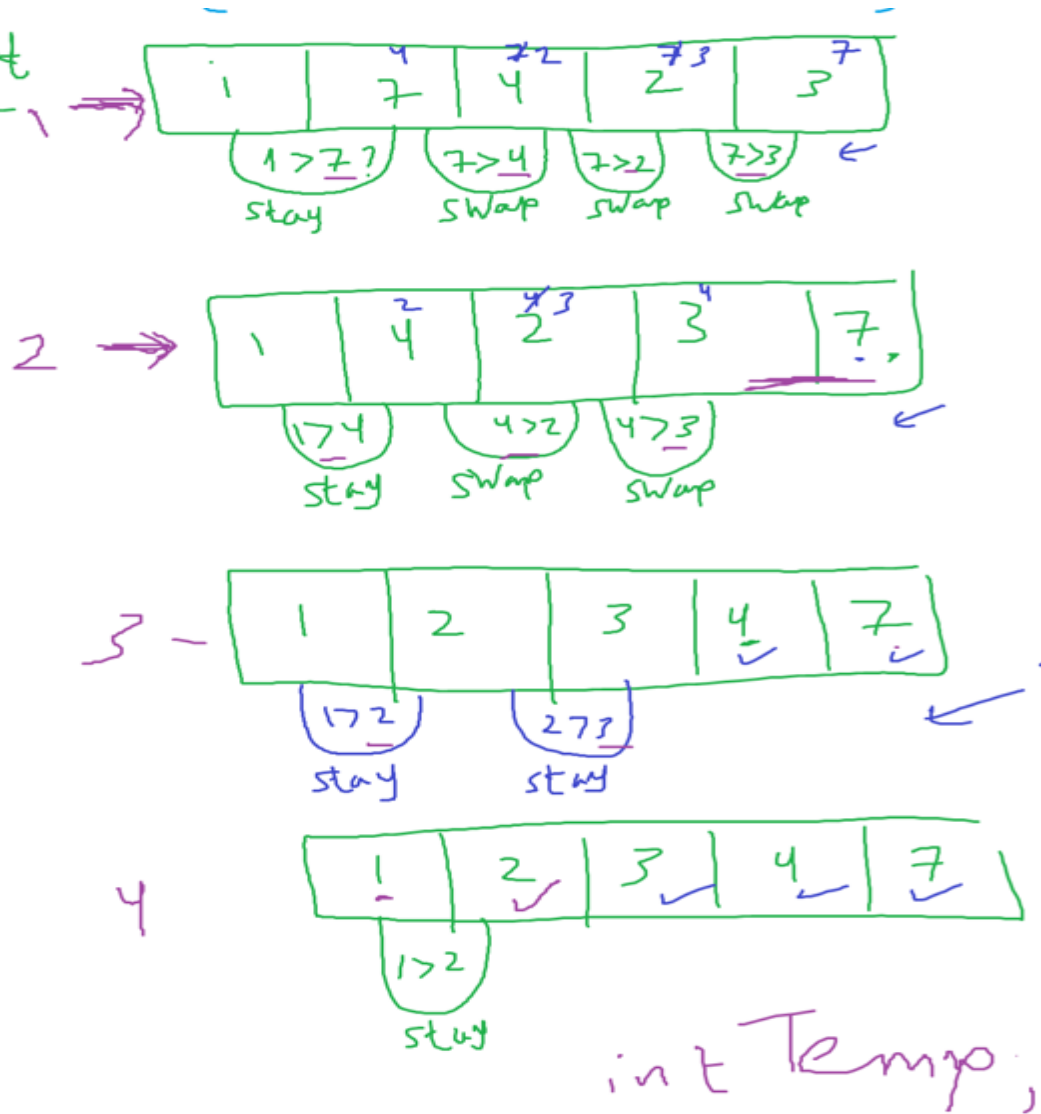
```

C:\Windows\System32\cmd.exe
mo:"project 1.c"
mo:a.exe
Enter element num 1 :5
Enter element num 2 :9
Enter element num 3 :8
Enter element num 4 :6
Enter element num 5 :7
Enter element num 6 :4
Enter element num 7 :2
Enter element num 8 :3
Enter element num 9 :1
Enter element num 10 :-5
the array is : -5 ,1 ,2 ,3 ,4 ,5 ,6 ,7 ,8 ,9 ,
mo:_

```

احنا ممكن نحسن الكود اكتر:

Bubble Sort



زي ما احنا ملاحظين في الرسمه ان لما بوصل للفه الي قبل الخيره بتكون ال array مترتبه وان اخر لفه دي ملهاش لزمه

(امشي بورقه وقلم علشان تبقي واضحه) فعلشان كده بنعمل عدد لفات اقل من ال size بواحد

نلاحظ انه في الترتيب انتقلت العناصر الأكبر الي اخر ال Array ده في حاله الترتيب التصاعدي والعكس بالعكس في الترتيب التنازلي

فلايوجد أي فائده من اعاده ترتيب العناصر الأخيره كما هو واضح في الرسم وبالتالي هنقلل عدد اللفات الي غير مفيده

وده هيكون من خلال ان في كل لفه بكون متأكد ان فيه عنصر اخذ مكانه في ال Array وبالتالي هنقلل عدد اللفات مع كل لفه بمقدار واحد

وال هو واضح ان ال i بتزيد بواحد في كل لفه وبتكون بصفر في اول لفه فهنستغلها

وبالتالي الكود بعد التعديل سيكون كده:

```
1  #include <stdio.h>
2  // #include "functions.h"
3  int main(void)
4  {
5      int temp;
6      int arr[10];
7      for(int i=0;i<10;i++)
8      {
9          printf("Enter element num %d :",i+1);
10         scanf("%d",&arr[i]);
11     }
12
13     for(int i=0;i<9;i++)
14     {
15         for(int y=0;y<9-i;y++)
16         {
17             if(arr[y]>arr[y+1])
18             {
19                 temp=arr[y];
20                 arr[y]=arr[y+1];
21                 arr[y+1]=temp;
22             }
23         }
24     }
25     printf("the array is : ");
26     for(int i=0;i<10;i++)
27     {
28         printf("%d ,",arr[i]);
29     }
30     return 0;
31 }
```

C:\Windows\System32\cmd.exe

mo: gcc "project 1.c"

mo: a.exe

Enter element num 1 :5

Enter element num 2 :8

Enter element num 3 :7

Enter element num 4 :4

Enter element num 5 :6

Enter element num 6 :9

Enter element num 7 :3

Enter element num 8 :2

Enter element num 9 :4

Enter element num 10 :1

the array is : 1 ,2 ,3 ,4 ,4 ,5 ,6 ,7 ,8 ,9 ,

mo: