TIKRIT UNIVERSITY
COLLEGE OF COMPUTER SCIENCE AND MATHEMATICS
DEPARTMENT OF COMPUTER SCIENCE

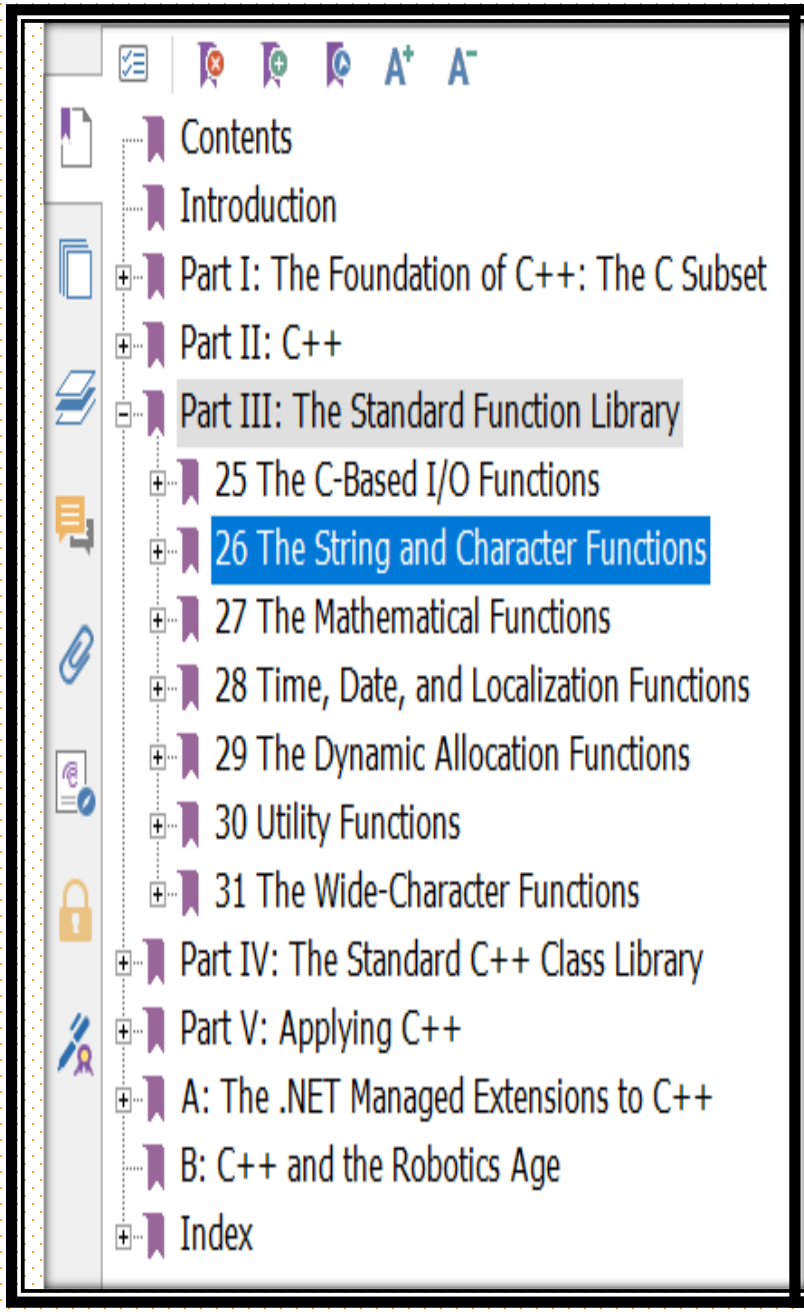
SUBJECT OF COMPILER (1)
DATE OF ISSUE: 2024 - 2025
CLASS: 3TH STAGE
SEMESTER 1
LAB-NO. : 1

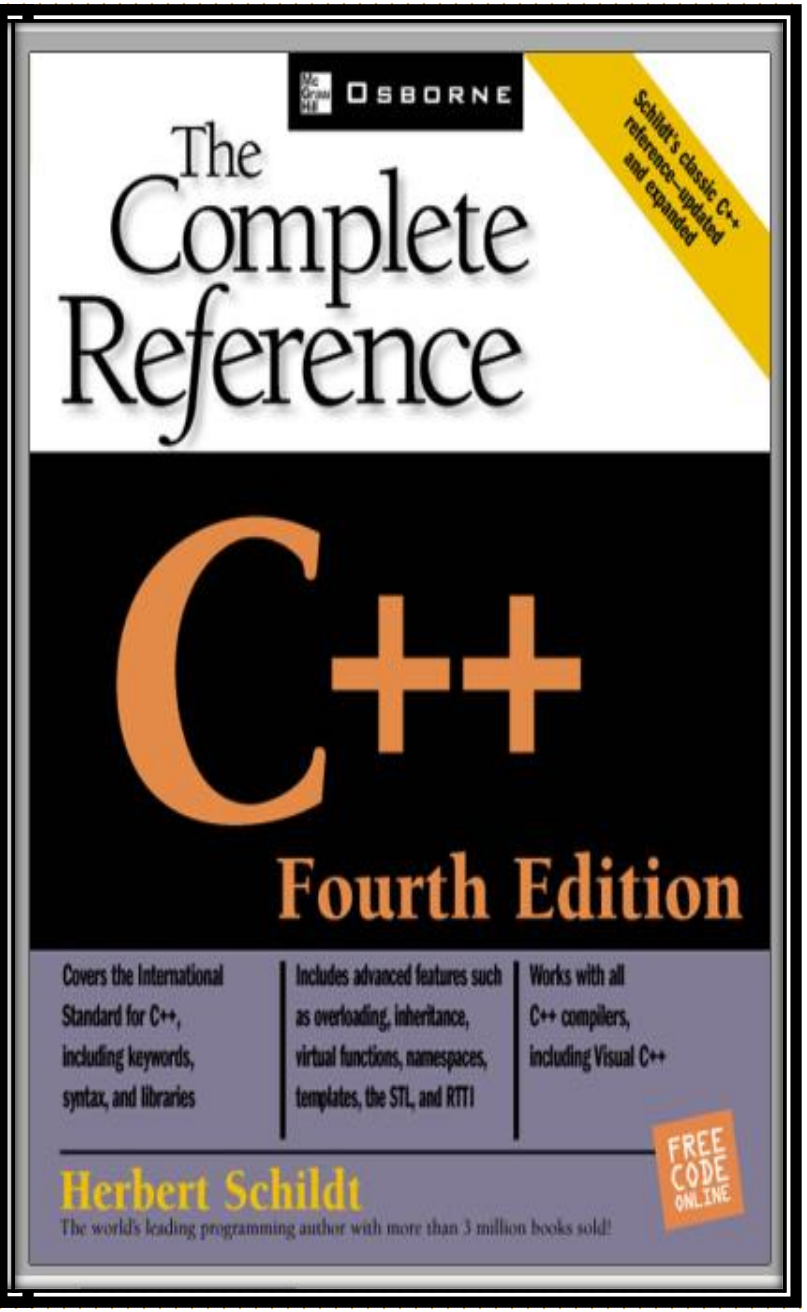
PREPARED BY

Lecturer: Mohanad Dawood Al-Roomi		Assistant Lecturer: Luay Ibrahim Klalif

The Sources of compiler ?

(2)





المحتويات

- **المقدمة:**
  - ✓ قراءة وتنسيب أو اسناد قيمة الى متغير نوعه البياني رمز char .
  - ✓ قراءة وتنسيب أو اسناد قيمة الى متغير نوعه البياني صحيح int .
  - ✓ قراءة وتنسيب أو اسناد قيمة الى متغير نوعه البياني حقيقي float .
- **مصفوفات الرموز في لغة : C & C++ (Character Arrays)**
  - ✓ مصفوفة ذات البعد الواحد.
  - ✓ مصفوفة ذات البعدين.
  - ✓ قراءة مصفوفات الرموز من لوحة المفاتيح في لغة C & C++
- **معالجة المصفوفات الرمزية غير المنتهية (Character Arrays of Null-Terminated)**
  1. دالة (strcpy( )) تنسيب أو اسناد أو نسخ قيمة الى مصفوفة خيوط رمزية.
  2. دالة (strcmp( )) مقارنة الخيوط الرمزية.
  3. دالة (strcat( )) دمج (الحاق) خيط رمزي مع خيط رمزي أخر.
  4. دالة (strlen( )) تعيد طول خيط رمزي.
  5. دالة (strchr( )) ترجع المؤشر الى تواجد الأول للـــ char في الخيط الرمزي.
  6. دالة (strrchr( )) ترجع المؤشر الى تواجد الاخير بايت في الخيط الرمزي.
  7. دالة (strstr( )) إرجاع مؤشر إلى أول ظهور لـ str2 في str1 .
- **معالجة السلاسل (الخيوط) الرمزية (The String Class)**

(4) قراءة وتنسيب أو اسناد قيمة الى متغير نوعه البياني رمز char

تعريف متغير نوعه البياني char في لغة C++ حجمة واحد بايت و اسناد قيمة رمز واحد اليه بشكل مباشر.

```
#include <iostream>
#include <cstring>
using namespace std;
int main()
{
    char ch = 'A';
    cout << " ch = " << ch << endl ;
    return 0;
}
```

RUN

```
ch = A
```

تعريف متغير نوعه البياني char في لغة C++ حجمة واحد بايت ثم اسناد قيمة رمز واحد اليه.

لاحظ: عند وضع رمز A بين علامتي اقتباس مفرد يتم اعتباره رمز char .

```
#include <iostream>
#include <cstring>
using namespace std;
int main()
{
    char ch ;
    ch = 'A';
    cout << " ch = " << ch << endl ;
    return 0;
}
```

RUN

```
ch = A
```

تعريف متغير نوعه البياني char في لغة C++ حجمة واحد بايت وتنسيب قيمه له ثم تنسيبه الى متغير ثاني.

```
#include <iostream>
#include <cstring>
using namespace std;
int main()
{
    char ch1 = 'A';
    cout << " ch1 = " <<ch1<<endl;
    char ch2 ;
    ch2 = ch1 ;
    cout << " ch2 = " <<ch2<<endl;
    return 0;
}
```

RUN

```
ch1 = A
ch2 = A
```

تعريف متغير نوعه البياني char في لغة C++ حجمة واحد بايت ثم قراءة هذا الرمز من خلال دوال القراءة ومن ثم ادخال قيمته من خلال target وتنسيبه الى متغير program ثاني.

```
#include <iostream>
#include <cstring>
using namespace std;
int main()
{
    char ch1 , ch2 ;
    cout<< " Enter your character = ";
    cin>> ch1;
    cout<< " ch1 = " <<ch1<<endl;
    ch2 = ch1 ;
    cout<< " ch2 = " <<ch2<<endl;
    return 0;
}
```

```
Enter your character = 5
ch1 = 5
ch2 = 5
```

لاحظ تم ادخال العدد (5) في هذه الحالة هل يتم اعتباره رمز char ام رقم umber ؟ اثبت ذلك؟ من خلال إضافة ايعازات على البرنامج؟

(5) قراءة وتنسيب أو اسناد قيمة الى متغير نوعه البياني رمز char

```cpp
#include <iostream>
#include <cstring>
using namespace std;
int main()
{
    char ch1 , ch2 ;
    cout<< " Enter your character = ";
    cin>> ch1;
    cout<< " ch1 = " <<ch1<<endl;
    ch2 = ch1 ;
    cout<< " ch2 = " <<ch2<<endl;
    cout<< " ch1 = " << 0 + ch1 <<endl;
    return 0;
}
```

لاحظ:
تم اعتباه رمز وليس حرف وذلك بسبب الإعلان عنه كرمز char وليس رقم صحيح int او حقيقي float .

RUN

```
Enter your character = 5
ch1 = 5
ch2 = 5
ch1 = 53
```

ملاحظة:
تعريف متغير نوعه البياني int في لغة C++ حجمة 2 بايت ثم اسناد قيمه رقميه اليه واجب على الطلاب.

ملاحظة:
تعريف متغير نوعه البياني float في لغة C++ حجمة 4 بايت ثم اسناد قيمه رقميه اليه واجب على الطلاب.

HW: أكتب برنامج يطلب من المستخدم إدخال أي رمز سواءً حرف أو غيره (من لوحة المفاتيح) ثم أطبع الرقم المقابل له في نظام ترميز الآسكي كود (ASCII Code) ؟

HW: أكتب برنامج يطلب من المستخدم إدخال أي رمزين سواءً حرف أو غير ذلك (من لوحة المفاتيح) ثم يرتبهما تصاعدياً وفق نظام ترميز الآسكي كود (ASCII Code) ؟

قراءة وتنسيب أو اسناد قيمة الى متغير نوعه البياني رمز char

```
1  #include <iostream>
2  #include <cstring>
3  using namespace std;
4  int main()
5  {
6      char ch ;
7      ch ='ABCD';
8      cout << " ch = " << ch << endl ;
9      return 0;
10 }
11
```

تعريف متغير نوعه البياني char في لغة C++ حجمة واحد بايت ثم اسناد قيمة اربع رموز اليه. لاحظ اخذ الرمز الأخير فقط.

خطأ مبرمج
Logical errors

RUN

ch = D

line7: warning: multi-character character constant. [-Wmultichar]. السطر 7: تحذير: ثابت متعدد الأحرف

line7: warning: overflow in implicit constant conversion. [-Woverflow] السطر 7: تجاوز في التحويل الثابت الضمني

```
1  #include <iostream>
2  #include <cstring>
3  using namespace std;
4  int main()
5  {
6      char ch = 'ABCD';
7      cout << " ch = " <<ch<<endl;
8      return 0;
9  }
```

تعريف متغير نوعه البياني char في لغة C++ حجمة واحد بايت و اسناد قيمة اربع رموز اليه بشكل مباشر.لاحظ اخذ الرمز الأخير فقط.

خطأ مبرمج
Logical errors

RUN

ch = D

line7: warning: multi-character character constant. [-Wmultichar]. السطر 7: تحذير: ثابت متعدد الأحرف

line7: warning: overflow in implicit constant conversion. [-Woverflow] السطر 7: تجاوز في التحويل الثابت الضمني

```
1  #include <iostream>
2  #include <cstring>
3  using namespace std;
4  int main()
5  {
6      char ch1 = 'A';
7      char ch2 = 'B';
8      char ch3 ;
9      ch3 = ch1 + ch2 ;
10     cout << " ch3 = " <<ch3<<endl;
11     return 0;
12 }
```

كيف يتم دمج متغيران نوعهما البياني char في لغة C++. لاحظ النتيجة كيف تفسر ذلك؟

خطأ مبرمج
Logical errors

RUN

ch3 = â

مصفوفات الرموز في لغة : C & C++ (Character Arrays)

❑ المصفوفة Array :

- ✓ المصفوفة هي مجموعة من المتغيرات من نفس النوع والتي يشار إليها من خلال إعطائها اسم معرف (Identifier).
- ✓ مثل المتغيرات الأخرى ، يجب التصريح عن المصفوفات بصراحةً حتى يتمكن المترجم من تخصيص مساحة لها في الذاكرة. هنا ، تحدد الكتابة النوع الأساسي للمصفوفة ، وهو نوع كل عنصر في المصفوفة ، ويحدد الحجم [Index] عدد العناصر التي ستحتويها المصفوفة.
- ✓ في C / C ++، تتكون المصفوفات من مواقع ذاكرة متجاورة .يتوافق العنوان الأدنى للذاكرة مع عنصرها الأول والذي يرقم بالرقم [0]، والعنوان الأعلى مع عنصرها الأخير والذي يرقم بالرقم [Index - 1] .
- ✓ يتم الوصول إلى أي عنصر عن طريق فهرسة (Index) لاسم المصفوفة .يتم ذلك عن طريق وضع فهرس العنصر بين أقواس مربعة بعد اسم المصفوفة.
- ✓ قد يكون للمصفوفات أبعاد من واحد إلى عدة أبعاد .المصفوفة الأكثر شيوعاً هي السلسلة المنتهية بقيمة خالية (موضوع دراستنا) ، والتي هي ببساطة مصفوفة من الرموز منتهية بصفر Null .
- ✓ اذا وجد المترجم النوع البياني char يقوم بحجز (1) بايت لكل عنصر في المصفوفة. واذا كان النوع البياني int يقوم بحجز (2) بايت لكل عنصر في المصفوفة. أما اذا كان النوع البياني float يقوم بحجز (4) بايت لكل عنصر في المصفوفة.
- ✓ في اللغات العليا يتم التعامل مع اسم المصفوفة ومن ثم موقع العنصر في المصفوفة للوصول الى محتوى ذلك العنصر.
- ✓ عند الإعلان عن مصفوفة لن يقوم المترجم بعد الحجز بإفراغ الخلايا من أي محتوى. لذلك يجب اسناد قيمة معلومة الى المصفوفة للتأكد من افراغ محتوى المصفوفة .

مصفوفات الرموز في لغة : C & C++ (Character Arrays)

1. مصفوفة ذات البعد الواحد نتعامل معها بأحدى الطرق التالية:
   A. كمتغير واحد نوعه البياني string.
   B. كمصفوفة أحادية أي نتعامل معها وقراءتها عنصر بعد اخر.

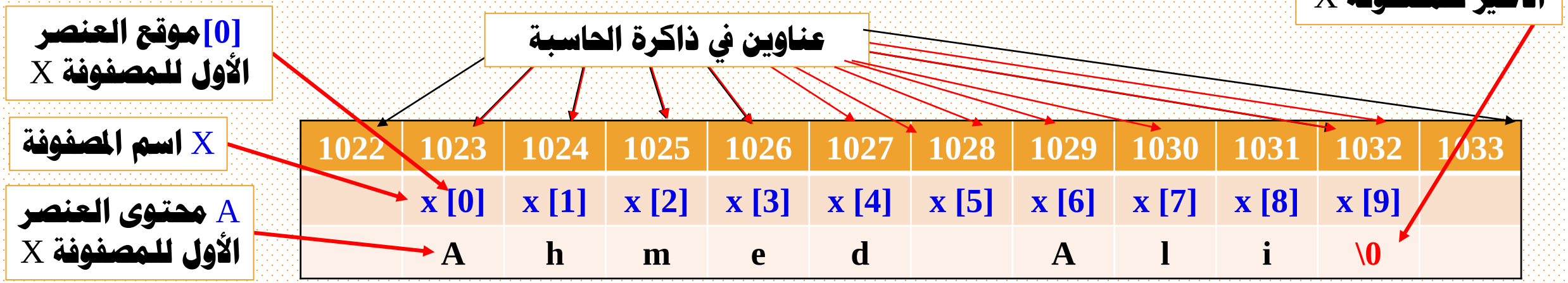
للإعلان عن مصفوفة احادية ; type var_name[size]

```
char X [10] = "Ahmed Ali";
```

```
char X [  ] = "Ahmed Ali";
```

```
char X [10] = {'A','h','m','e','d',' ','A','l','i','\0'};
```

```
char X [  ] = {'A','h','m','e','d',' ','A','l','i','\0'};
```



يتم الوصول إلى أي عنصر عن طريق index اسم المصفوفة .يتم ذلك عن طريق وضع index العنصر بين أقواس مربعة بعد اسم المصفوفة .فمثلا،.في C / C ++، تحتوي جميع المصفوفات على الرقم [0] كمؤشر لعنصرها الأول .لذلك ، عندما تكتب

char X [10] ;

أنت تعلن عن مصفوفة رموز بها عشرة عناصر ، من X [0] إلى X[9] .

- ✓ **في المثال السابق: اعلان عن متغير مصفوفة أحادية نوع رموز** char **مع اسناد قيمه رمزية لها. (تسمح** C++ **باستخدام رمز المساوات لإسناد قيمة رمزية الى متغير رمزي نوع مصفوفة وقت الإعلان فقط). أما في البرنامج الرئيسي فيجب ان نستخدم دالة** strcpy() **لإسناد قيمة رمزية الى مصفوفة.**
- ✓ **المترجم بالتعاون مع نظام التشغيل يبحث عن عشرة مواقع (حسب النوع البياني) متسلسلة غير مستخدمة من قبل البرنامج في الذاكرة ليقوم بحجزها تحت اسم المصفوفة** X.
- ✓ **في لغة** C++ **فأن السلاسل الرمزية عي عبارة عن مصفوفة للرموز تنتهي بالرمز (null) (رمز النهاية) حيث يمثل نهاية السلسلة الرمزية، بالإمكان ان تعلن وتبتدأ السلاسل الرمزية كما تفعل بالضبط مع مصفوفات البيانات من الاعداد الصحيحة والحقيقية، مثال:**

```
char Greeting[ ] = { 'H', 'e', 'l', 'l', 'o', ' ', 'W', 'o', 'r', 'l', 'd', '\0' };
```

لاحظ ان الحرف الاخير هو ('\0') (null) ما بعد الشرطة المعكوسة هو صفر.

- ✓ **لغة** C++ **توفر إمكانية اختصار الطريقة أعلاه والتي تعتمد على ادخال رمز بعد الأخر، وكما يأتي:**

```
char Greeting[ ] = "Hello world";
```

حيث ان هذه القاعدة توفر شيئين:

- **فبدلاً من استخدام الحاصرات المفردة المفصولة بالفوارز والمحاطة بالأقواس فانك ستستخدم الحاصرات المزدوجة بدون فوارز واقواس.**
- **عدم الحاجة لاضافة حرف النهاية لان المترجم سيضيفه عوضاً عنك.**

- ✓ **هنا حجم المصفوفة يساوي** (12 Byte) **وذلك لان كلمة** (Hello) **تحتاج الى خمس بايتات، وفراغ واحد يحتاج بايت واحد، وكلمة** (world) **تحتاج الى خمس بايتات، واخيراً بايت واحد لحرف النهاية.**
- ✓ **ملاحظة: عندما يتم ابتداء القيم سوف تسند لعناصر المصفوفة،** C++ **يسمح بإمكانية ترك الاقواس المربعة فارغة [ ]، في هذه الحالة فان المترجم سيفرض حجم للمصفوفة يطابق عدد القيم الموجودة بين الاقواس المتوسطة. في المثال هنا سيحدد المترجم عدد العناصر هي** 12 .

مصفوفات الرموز في لغة : C & C++ (Character Arrays)

2. مصفوفة ذات البعدين (Two-Dimensional Arrays)

نتعامل معها كمصفوفة بعد واحد (الطول لا يدخل في التعامل)


```
char X [10][5] ;
```

- ✓ لإنشاء مصفوفة من السلاسل المنتهية بصفر، استخدم مصفوفة أحرف ثنائية الأبعاد .يحدد حجم الفهرس الأيسر عدد السلاسل (الصفوف) ويحدد حجم الفهرس الأيمن الحد الأقصى لطول كل سلسلة (العِمود) .
- ✓ مثال يعلن الكود التالي عن مصفوفة من 30 سلسلة ، كل منها بحد أقصى 79حرفا ، بالإضافة إلى حرف النهاية الفارغ.

```
char str_array [30] [80];
```

- ✓ من السهل الوصول إلى سلسلة فردية (الصف) :ما عليك سوى تحديد الفهرس الأيسر فقط.على سبيل المثال، استدعاءات العبارات التالية:

```
gets(str_array[2]);
```

سوف تحصل على سلسلة الثالثة (الصف الثالث) من المصفوفة str_array .

3.المصفوفات متعددة الأبعاد (Multidimensional Arrays)

تسمح لغة C / C ++ للمصفوفات بأكثر من بعدين. الشكل العام لإعلان مصفوفة متعددة الأبعاد هو

```
type name[Size1][Size2][Size3]. . .[SizeN];
```

مصفوفات الرموز في لغة : C & C++ (Character Arrays)

```cpp
#include <iostream>
#include <string>
using namespace std;
int main()
{
    char A = 'a';
    cout << "Character A = " << A << endl;

    char B[2] = "a";
    cout << "First element of character Array  B = " << B[0] << endl;
    cout << "Second element of character Array B = " << B[1] << endl;
    cout << "All elements of character Array   B = " << B    << endl;

    string C ("a");
    cout << "String C = " << C << endl;
    return 0;
}
```

الإعلان عن متغير ثابت الرمز (نوع char)

الإعلان عن متغير نوع مصفوفة رموز char

الإعلان عن متغير نوع خيط رموز نوعه String

```
Character A = a
First element of character Array  B = a
Second element of character Array B =
All elements of character Array   B = a
String C = a
```

ملاحظة: يجب ألا تخلط بين ثابت الرمز Character ومصفوفة الرموز غير المنتهية (Character arrays of Null-Terminated) والسلاسل الرمزية (String Class).

- ✓ يتم وضع ثابت الرمز واحد في علامات اقتباس مفردة كما هو الحال في 'a' .
- ✓ بينما يتم وضع مصفوفة الرموز في علامات اقتباس مزدوجة كما هو الحال في "a" هنا يتم اعتبارها مصفوفة رموز تحتوي على رمزين الرمز (a) ظاهر والرمز Null مخفي هو (\0).
- ✓ وكذلك يتم وضع سلسلة الرموز في علامات اقتباس مزدوجة كما هو الحال في "a" هنا يتم اعتبارها سلسلة رموز تحتوي على رمز واحد فقط.

مصفوفات الرموز في لغة : C & C++ (Character Arrays)

```cpp
#include <iostream>
#include <cstring>
using namespace std;
int main()
{
   char X[12]= {'H','e','l','l','o',' ','W','o','r','l','d','\0'};
   char Y[12]= {'H','e','l','l','o',' ','W','o','r','l','d'};
   cout << "length Array X= " << strlen(X) << endl;
   cout << "length Array Y= " << strlen(Y) << endl;
   for (int i=0 ; i<20 ; i++)
     {
       cout <<"      X["<<i<<"]= " << X[i];
       cout <<"  ,   Y["<<i<<"]= " << Y[i]<< endl;
     }
    return 0;
}
```

```
length Array X= 11
length Array Y= 11
      X[0]= H  ,   Y[0]= H
      X[1]= e  ,   Y[1]= e
      X[2]= l  ,   Y[2]= l
      X[3]= l  ,   Y[3]= l
      X[4]= o  ,   Y[4]= o
      X[5]=    ,   Y[5]=
      X[6]= W  ,   Y[6]= W
      X[7]= o  ,   Y[7]= o
      X[8]= r  ,   Y[8]= r
      X[9]= l  ,   Y[9]= l
      X[10]= d  ,   Y[10]= d
      X[11]=    ,   Y[11]=
      X[12]= ♀  ,   Y[12]= H
      X[13]=    ,   Y[13]= e
      X[14]=    ,   Y[14]= l
      X[15]=    ,   Y[15]= l
      X[16]= -  ,   Y[16]= o
      X[17]= o  ,   Y[17]=
      X[18]= ↑  ,   Y[18]= W
      X[19]= v  ,   Y[19]= o
```


```
length Array X= 11
length Array Y= 11
      X[0]= H  ,   Y[0]= H
      X[1]= e  ,   Y[1]= e
      X[2]= l  ,   Y[2]= l
      X[3]= l  ,   Y[3]= l
      X[4]= o  ,   Y[4]= o
      X[5]=    ,   Y[5]=
      X[6]= W  ,   Y[6]= W
      X[7]= o  ,   Y[7]= o
      X[8]= r  ,   Y[8]= r
      X[9]= l  ,   Y[9]= l
      X[10]= d  ,   Y[10]= d
      X[11]=    ,   Y[11]=
      X[12]= ♀  ,   Y[12]= H
      X[13]=    ,   Y[13]= e
      X[14]=    ,   Y[14]= l
      X[15]=    ,   Y[15]= l
      X[16]= -  ,   Y[16]= o
      X[17]= o  ,   Y[17]=
      X[18]= ↑  ,   Y[18]= W
      X[19]= v  ,   Y[19]= o
```


```cpp
#include <iostream>
#include <cstring>
using namespace std;
int main()
{
   char X[12]= "Hello world";
   char Y[11]= "Hello world";
   cout << "length Array X= " << strlen(X) << endl;
   cout << "length Array Y= " << strlen(Y) << endl;
   for (int i=0 ; i<20 ; i++)
     {
       cout <<"      X["<<i<<"]= " << X[i];
       cout <<"  ,   Y["<<i<<"]= " << Y[i]<< endl;
     }
    return 0;
}
```

خطأ معاني
Semantic errors


```cpp
#include <iostream>
#include <cstring>
using namespace std;
int main()
{
   char X[12]= "Hello world";
   char Y[12]= "Hello world";
   cout << "length Array X= " << strlen(X) << endl;
   cout << "length Array Y= " << strlen(Y) << endl;
   for (int i=0 ; i<20 ; i++)
     {
       cout <<"      X["<<i<<"]= " << X[i];
       cout <<"  ,   Y["<<i<<"]= " << Y[i]<< endl;
     }
    return 0;
}
```

مصفوفات الرموز في لغة : C & C++ (Character Arrays)

```cpp
#include <iostream>
#include <cstring>
using namespace std;
int main()
{
    char X[ ]= "Hello world";
    char Y[ ]= {'H','e','l','l','o',' ','W','o','r','l','d','\0'};
    cout << "length Array X= " << strlen(X) << endl;
    cout << "length Array Y= " << strlen(Y) << endl;
    for (int i=0 ; i<20 ; i++)
      {
        cout <<"       X["<<i<<"]= " << X[i];
        cout <<"  ,   Y["<<i<<"]= " << Y[i]<< endl;
      }
     return 0;
}
```

```
length Array X= 11
length Array Y= 11
      X[0]= H  ,   Y[0]= H
      X[1]= e  ,   Y[1]= e
      X[2]= l  ,   Y[2]= l
      X[3]= l  ,   Y[3]= l
      X[4]= o  ,   Y[4]= o
      X[5]=    ,   Y[5]=
      X[6]= W  ,   Y[6]= W
      X[7]= o  ,   Y[7]= o
      X[8]= r  ,   Y[8]= r
      X[9]= l  ,   Y[9]= l
      X[10]= d ,   Y[10]= d
      X[11]=   ,   Y[11]=
      X[12]= ♀ ,   Y[12]= H
      X[13]=   ,   Y[13]= e
      X[14]=   ,   Y[14]= l
      X[15]=   ,   Y[15]= l
      X[16]= - ,   Y[16]= o
      X[17]= o ,   Y[17]=
      X[18]= ↑ ,   Y[18]= W
      X[19]= v ,   Y[19]= o
```

```cpp
#include <iostream>
#include <cstring>
using namespace std;
int main()
{
    char XX[10];
    char Y[10][10];
    cout << "length Array XX= " << strlen(XX) << endl;
    for (int i=0 ; i<10 ; i++)
      {
        cout <<"       XX["<<i<<"]= " << XX[i];
        cout <<"  ,   Y["<<i<<"]= " << Y[i]<< endl;
      }
     return 0;
}
```

خطأ مبرمج
Logical errors

```
length Array XX= 13
      XX[0]= ↑  ,   Y[0]=
      XX[1]= v  ,   Y[1]= ⌂fA
      XX[2]= «  ,   Y[2]= ⌂
      XX[3]= ⌂  ,   Y[3]= ⁿ∟⌂
      XX[4]= c  ,   Y[4]= ↓v∟C▲v▄▀i
  ,   Y[5]= ⌂d↓v
      XX[6]= ■  ,   Y[6]= ↑v#n↑v‖é↕{⌂↕@
      XX[7]=    ,   Y[7]= ⌂↕@
      XX[8]=    ,   Y[8]=
      XX[9]=    ,   Y[9]= ⌂■i
```

قراءة سلسلة رمزية من لوحة المفاتيح في لغة C & C++

```
#include <iostream>
#include <cstring>
using namespace std;
int main()
{
    char X[50];
    cout <<"Enter your full Name:";
    cin >> X ;
    int L = strlen(X);
    cout <<"L= " << L << endl;
    cout <<"X= " <<X << endl;
    return 0;
}
```

```
Enter your full Name:Mohanad Dawood Salman
L= 7
X= Mohanad
```

1 . قراءة سلسلة رمزية باستخدام دالة cin >> .

cin >> *array-name* ;

مشكلة الأولى: التي سوف تحدث وهي عند الضغط على التنفيذ وظهور نافذه (target program) وإدخال المستخدم عبارة تحتوي على فراغات سوف يأخذ البرنامج فقط المدخلات التي قبل الفراغات والسبب في ذلك ان العامل (>>) توقف عملية قراءة السلسلة الرمزية عند ورود اول رمز لفضاءات الفراغ (whitespace). رمز فضاء الفراغات يتضمن (space) (tabs) (newlines).

المشكلة الثانية: العامل (>>) أو الدالة (gets()) لا يوفران فحص أو ضبط لحدود المصفوفة. لذلك اذا ما ادخل المستخدم سلسلة رمزية أطول من حجم المصفوفة، فأن المصفوفة ستكتب فوق العناصر اللاحقة أي بعد ان تتجاوز حدود المصفوفة، وهذا سيجعل كلتا طريقتي قراءة السلسلة خطرة على محتويات الذاكرة.

2. الدالة gets() تستخدم ملف الصديره

```
# include <cstdio>
gets(array-name);
```

قراءة سلسلة رمزية من لوحة المفاتيح في لغة C & C++

3.الدالة cin.getline() تستخدم ملف الصديره

```
# include <cstdio>
cin.getline( string_var ،max_characters + 1) ;
```

الدالة العضو getline () من الممكن ان تستخدم لقراءة سطر من المدخلات ووضع رموز السلسلة الرمزية على هذا السطر بمتغير سلاسل حرفية. الصيغة القواعدية

سطر واحد من المدخلات يقرأ، والناتج والذي هو سلسلة رمزية يوضع في متغير سلاسل حرفية، فاذا كان السطر اكبر من الحجم المحدد او طول (max_characters+1) عندها فقط اول عدد من الحروف والتي تساوي (max_characters) على السطر سوف تقرأ. ان اضافة الرقم واحد ضروري لان السلاسل الرمزية في C دائما تنتهي برمز النهاية فراغ (\0). مثال

```
char one_line [80] ;
cin.getline (one_line, 80 ) ;
```

- برنامج لقراءة سلسلة رمزية باستخدام الدالة cin.getline()

```
#include <iostream>
#include <cstring>
using namespace std;
int main()
  {
    char buffer[80];
    do
      {
        cout << "Enter a string up to 80 characters: ";
        cin.getline(buffer, 80);
        cout << "Your string is " << strlen(buffer);
        cout << " characters long." <<endl;
      }
    while (strlen(buffer));
    cout << "\nDone." << endl;
  return 0;
}
```

```
Enter a string up to 80 characters: ABCD EFGH
Your string is 9 characters long.
Enter a string up to 80 characters: IJKL MNOPQ
Your string is 10 characters long.
Enter a string up to 80 characters:
Your string is 0 characters long.

Done.
```

```cpp
#include <iostream>
#include <cstring>
using namespace std;
const int MAX = 2000 ;
char str[MAX];//متغير سلاسل رمزية
int main()
{
    cout << "\n Enter a string:\n";
    cin.get(str, MAX , '$') ; // منتهية بالرمز$
    cout << "You entered:\n" << str << endl;
 return 0;
}
```

```
 Enter a string:
My name is Ahmed Khaled
I live in Salah El-Din
I am a student at the College of
Computer Science and Mathematics$
You entered:
My name is Ahmed Khaled
I live in Salah El-Din
I am a student at the College of
Computer Science and Mathematics
```

3. الدالة cin.get() لقراءة اسطر متعددة وتستخدم ملف الصديره

```
# include <cstdio>
cin.get( string_var ،max_characters + 1) ;
```

7.4.3 قراءة اسطر متعددة Reading Multiple Lines

ربما اصبحت الان قادرا على حل مشكلة قراءة سلسلة رمزية تحتوي على فراغات ضمنية، ولكن ماذا عن السلاسل الرمزية مع اسطر متعددة؟ هذا يتم باستخدام الدالة (cin.get()) التي ستساعد في هذه الحالة، وسيتم الاستعانة باستخدام معامل ثالث. هذا المعامل يحدد الرمز الذي سيخبر الدالة بايقاف القراءة. القيمة الافتراضية لهذا المعامل هي الرمز ('\n')، ولكن اذا استدعيت الدالة مع بعض الرموز الاخرى، فان القيمة الافتراضية سيتم تجاوزها بالرموز المحددة.

- برنامج يقرأ عدد من اسطر السلاسل الرمزية تنتهي بالرمز '$'

الان بامكانك طباعة اي عدد من الاسطر المدخلة التي تريدها. الدالة ستستمر بقبول الرموز لغاية ادخال رمز النهاية (او لغاية تجاوز حجم المصفوفة). تذكر، لازال وجوبا عليك ان تضغط زر الادخال (Enter) بعد طباعة الرمز ($). مخرجات البرنامج هي

في هذا البرنامج ستقوم بانهاء كل سطر بالضغط على زر الادخال، ولكن البرنامج سيستمر بقبول المدخلات لحين ان تقوم بادخال الرمز $.

قراءة سلسلة رمزية من لوحة المفاتيح في لغة C & C++

برنامج لقراءة مصفوفة احرف وطباعتها

```
//Example 5.8
#include <iostream>
using namespace std;

int main()
{
char buffer[80];
cout << "Enter the string: ";
cin >> buffer;
cout << "Here's the buffer: " << buffer << endl;
return 0;
}
```

مخرجات البرنامج 5.8:

```
Enter the string: Hello World
Here's the buffer: Hello
```

لاحظ مدخلات ومخرجات البرنامج 5.8 (والذي يجب ان تتأكد بعدم وضع احرف اكثر من الحجم المحدد) حيث ان المصفوفة معرفة بحجم (80) حرف اي بامكانك ان تضع (79) حرف لان الحرف الاخير يمثل حرف النهاية، ولكن من الملاحظ هنا وجود مشكلتين هما:

- اولا يجب ان تتاكد بعدم ادخال اكثر من (79) حرف لان ذلك سيؤدي الى وضع قيم خارج مدى المصفوفة.
- اذا ما قمت بادخال فراغ فان المترجم سيترجمة على انه نهاية السلسلة ويتوقف عن اسناد الاحرف التالية للفراغ الى المتغير الرمزي (buffer) كما في ناتج البرنامج.

لحل هاتين المشكلتين عليك استدعاء دالة خاصة هي (cin:get ()). هذه الدالة تأخذ ثلاثة وسائط:

* المتغير الذي يجب وضع الحروف به.
* الحد الاقصى لعدد الحروف الواجب وضعها.
* اشارة النهاية (اشارة النهاية الافتراضية هي سطر جديد).

- برنامج لقراءة وطباعة مصفوفة من الاحرف باستخدام cin.get()

```
//Example 5.9
#include <iostream>
using namespace std;

int main()
{
char buffer[80];
cout << "Enter the string: ";
cin.get(buffer, 79); // get up to 79 or newline
cout << "Here's the buffer: " << buffer << endl;
return 0;
}
```

مخرجات البرنامج 5.9:

```
Enter the string: Hello World
Here's the buffer: Hello World
```

في البرنامج 5.9 تم استخدام الامر (cin:get()) مع المتغيرات التالية، (buffer) وهو المتغير الذي ستضع فيه السلسلة الرمزية (طبعا هو مصفوفة من الاحرف) والرقم (79) والذي يمثل الحد الاعلى للحروف في السلسلة الرمزية، ولم يذكر الوسيط الثالث حيث سيفرضة المترجم (سطر جديد New Line))، هذا الايعاز سيسمح باسناد حروف الى المتغير (buffer) لغاية (79) او لغاية ادخال سطر جديد. وفي هذه الحالة فان حرف النهاية سيتم وضعة في نهاية السلسلة عندما تدخل (79) حرف اما في المثال 5.9 فلا حاجة لتوفير حرف النهاية وذلك لان القيمة الافتراضية (سطر جديد) ستكون كافية.

(18) معالجة المصفوفات الرمزية غير المنتهية (Arrays Null-Terminated Strings)

1. دالة (strcpy) تنسيب أو اسناد أو نسخ قيمة الى مصفوفة خيوط رمزية

```
#include <cstring>
char *strcpy(char *str1 , const char * str2);
```

- ✓ The strcpy( ) function copies the contents of *str2* into *str1*.
- ✓ *str2* must be a pointer to a null-terminated string.
- ✓ The strcpy( ) function returns a pointer to *str1*.
- ✓ If *str1* and *str2* overlap, the behavior of strcpy( ) is undefined.

- ✓ حيث نضمن ملف الصديره <string.h> الموجود في لغة C أو نضمن ملف الصديره <cstring> الموجود في لغة C++ لاستدعاء الدالة strcpy() .
- ✓ تنسخ الدالة strcpy() محتويات str2 إلى str1 .
- ✓ يجب ان تكون المصفوفة (str1) التي ستنقل لها السلسلة الرمزية كبيرة بدرجة كافية لاستيعاب السلسلة التي في str2 .
- ✓ أما اذا لم تكن كافية، فأن المصفوفة str1 سيتم تجاوزها أي سيتم الخزن الى ما بعد حجم المصفوفة، والتي ربما تؤدي الى تدمير البرنامج أو معلومات أخرى في الذاكرة. لأنها سوف تكتب على أماكن أخرى في الذاكرة غير مخصصة للمصفوفة.

char *str تشير الى متغير فقط نوعه البياني مصفوفة char

const char * str تشير الى ثابت أو متغير نوعه البياني مصفوفة char

- ❑ strcpy(variable1 , string)

  Ex: strcpy(X , "Good Morning");

- ❑ أما ان يتم نقل خيط رمزي الى متغير يتم تعريفه على انه مصفوفة نوع رموز.

- ❑ strcpy(variable1 , variable2)

  Ex: strcpy(X , Y);

- ❑ أو يتم نقل محتويات متغير نوعه البياني مصفوفة رموز الى متغير ثاني نوعه البياني مصفوفة رموز ايضاً.

1. دالة (strcpy) تنسيب أو اسناد أو نسخ قيمة الى مصفوفة خيوط رمزية

مثال(1): اكتب برنامج يقوم بأسناد خيط رمزي الى متغير نوعه مصفوفة رموز char ليس خلال الإعلان وانما خلال ايعازات البرنامج.

```
1  #include <iostream>
2  #include <cstring>
3  using namespace std;
4  int main()
5  {
6      char X[20];
7      strcpy(X , "Good Morning");
8      cout << X << endl ;
9      return 0;
10 }
```

RUN

Good Morning

مثال(2): اكتب برنامج يقوم بأسناد خيط رمزي الى متغير نوعه مصفوفة رموز char اثناء الإعلان عن المصفوفة.

```
1  #include <iostream>
2  #include <cstring>
3  using namespace std;
4  int main()
5  {
6      char X[20]= "Good Morning";
7      cout << X << endl ;
8      return 0;
9  }
```

مقبولة في لغة C and C++

RUN

Good Morning

مثال(3): اكتب برنامج يقوم بأسناد خيط رمزي الى متغير نوعه مصفوفة رموز char ليس خلال الإعلان وانما خلال ايعازات البرنامج.

```
1  #include <iostream>
2  #include <cstring>
3  using namespace std;
4  int main()
5  {
6      char X[20];
7      X = "Good Morning";
8      cout << X << endl ;
9      return 0;
10 }
```

خطأ قواعدي (نحوي) Syntactic errors

مقبولة في لغة باسكال
لكنها غير مقبولة في لغة C and C++

RUN

لاحظ العبارة والأرقام (13) و (20)

Message:
In function 'int main()':

|Line7| error: incompatible types in assignment of 'const char [13]' to 'char [20]'|

Build failed: 1 error(s), 0 warning(s) (0 minute(s), 0 second(s))

1. دالة (strcpy) تنسيب أو اسناد أو نسخ قيمة الى مصفوفة خيوط رمزية

```
1   #include <iostream>
2   #include <cstring>
3   using namespace std;
4   int main()
5   {
6       char X[12];
7       X = "Good Morning";
8       cout << X << endl ;
9       return 0;
10  }
```

مثال(4): اكتب برنامج يقوم بأسناد خيط رمزي الى متغير نوعه مصفوفة رموز char ليس خلال الإعلان وانما خلال ايعازات البرنامج.

خطأ قواعدي (نحوي) Syntactic errors

لاحظ العبارة والأرقام (12) و (20)

```
RUN
Message:
In function 'int main()':

|Line7| error: incompatible types in assignment
                of 'const char [12]' to 'char [20]'|

Build failed: 1 error(s), 0 warning(s)
              (0 minute(s), 0 second(s))
```

```
1   #include <iostream>
2   #include <cstring>
3   using namespace std;
4   int main()
5   {
6       char X[13];
7       X = "Good Morning";
8       cout << X << endl ;
9       return 0;
10  }
```

مثال(5): اكتب برنامج يقوم بأسناد خيط رمزي الى متغير نوعه مصفوفة رموز char ليس خلال الإعلان وانما خلال ايعازات البرنامج.

خطأ قواعدي (نحوي) Syntactic errors

لاحظ العبارة تطابق ولا توجد الأرقام

```
RUN
Message:
In function 'int main()':

|Line7| error: error: invalid array assignment

Build failed: 1 error(s), 0 warning(s)
              (0 minute(s), 0 second(s))
```

1. دالة (strcpy) تنسيب أو اسناد أو نسخ قيمة الى مصفوفة خيوط رمزية

```
1  #include <iostream>
2  #include <cstring>
3  using namespace std;
4  int main()
5  {
6      char X[12]= "Good Morning";
7      cout << X << endl ;
8      return 0;
9  }
```

مثال(6): اكتب برنامج يقوم بأسناد خيط رمزي الى متغير نوعه مصفوفة رموز char ليس خلال الإعلان وانما خلال ايعازات البرنامج.

خطأ معاني
Semantic errors

لاحظ العبارة

RUN

Message:

In function 'int main()':

|Line6| error: initializer-string for array of chars is too long.

Build failed: 1 error(s), 0 warning(s) (0 minute(s), 0 second(s))

```
1  #include <iostream>
2  #include <cstring>
3  using namespace std;
4
5  int main()
6  {
7      char X[12];
8      char Y[12];
9      strcpy(X , "Good morning");
10     cout << "length line X= " << strlen(X) << endl;
11     cout << "X= " << X <<endl;
12     strcpy(Y , X);
13     cout << "length line Y= " << strlen(Y) << endl;
14     cout << "Y= " << Y << endl;
15     cout << "X= " << X <<endl;
16     return 0;
17 }
```

```
length line X= 12
X= Good morning
length line Y= 12
Y= Good morning
X=
```

لاحظ النتيجة؟ ولماذا لم تطبع قيمة X النهائية ؟

لاحظ حجم المصفوفة التي تم حجزها اقل من الخيط الرمزي؟

```
1  #include <iostream>
2  #include <cstring>
3  using namespace std;
4
5  int main()
6  {
7      char X[13];
8      char Y[13];
9      strcpy(X , "Good morning");
10     cout << "length line X= " << strlen(X) << endl;
11     cout << "X= " << X <<endl;
12     strcpy(Y , X);
13     cout << "length line Y= " << strlen(Y) << endl;
14     cout << "Y= " << Y << endl;
15     cout << "X= " << X <<endl;
16     return 0;
17 }
```

```
length line X= 12
X= Good morning
length line Y= 12
Y= Good morning
X= Good morning
```

لاحظ تم طبع قيمة X النهائية ؟

1. دالة (strcpy) تنسيب أو اسناد أو نسخ قيمة الى مصفوفة خيوط رمزية

مثال (7): اكتب برنامج يقوم بتنسيب محتوى متغير خيط رمزي الى متغير ثاني؟

```
1  #include <iostream>
2  #include <cstring>
3  using namespace std;
4  int main()
5  {
6      char X[20] , Y[20];
7      strcpy(X , "Good Morning");
8      strcpy(Y , "Good Afternoon");
9      cout<< "X first time = " << X << endl ;
10     cout<< "Y first time = " << Y << endl ;
11     strcpy(Y , X);
12     cout<< "X second time = " << X << endl ;
13     cout<< "Y second time = " << Y << endl ;
14     return 0;
15 }
```

RUN

```
X first time = Good Morning
Y first time = Good Afternoon
X second time = Good Morning
Y second time = Good Morning
```

لاحظ: النتائج الأولية لـ X و Y والنتائج الثانوية ما تعليل ذلك؟

مثال (8): اكتب برنامج يقوم بتنسيب محتوى متغير خيط رمزي الى متغير ثاني؟ بشكل احترافي.

```
1  #include <iostream>
2  #include <cstring>
3  using namespace std;
4  int main()
5  {
6      char X[20] , Y[20] ;
7      cout<< "X first time = " << X << endl ;
8      cout<< "Y first time = " << Y << endl ;
9      strcpy(Y ,strcpy(X , "Good Morning"));
10     cout<< "X second time = " << X << endl ;
11     cout<< "Y second time = " << Y << endl ;
12     return 0;
13 }
```

RUN

```
X first time = ╠ i
Y first time = ⍰↕@
X second time = Good Morning
Y second time = Good Morning
```

لاحظ: يمكن استخدام هذا الصيغة حيث تعمل الدالة strcpy(X , "Good Morning") في البداية على اسناد الخيط الرمزي "Good Morning" الى المتغير X ثم تعمل الدالة strcpy(Y , X) على اسناد محتوى المتغير X الى المتغير Y وحسب اسبقية ما اداخل القوس.

1. دالة (strcpy) تنسيب أو اسناد أو نسخ قيمة الى مصفوفة خيوط رمزية

مثال(9): اكتب برنامج يقوم المستخدم بإدخال الحروف الخمسة الأولى من اللغة الإنكليزية وخزنها في المصفوفة A ومن ثم نقلها الى المصفوفة B مع طباعة عناصرها ؟

```
#include <iostream>
#include <cstring>
using namespace std;

int main()
{
    char A[5];
    char B[5];
    cout << "enter string for A" << endl;
    cin.getline(A,sizeof(A)+1);
    cout << " All elements of   A    = " << A << endl;
    cout << " First element of  A[0] = " << A[0] << endl;
    cout << " Second element of A[1] = " << A[1] << endl;
    cout << " Third element of  A[2] = " << A[2] << endl;
    cout << " Fourth element of A[3] = " << A[3] << endl;
    cout << " Fifth element of  A[4] = " << A[4] << endl;
    strcpy(B,A);
    cout << " All elements of   B    = " << B << endl;
    cout << " First element of  B[0] = " << B[0] << endl;
    cout << " Second element of B[1] = " << B[1] << endl;
    cout << " Third element of  B[2] = " << B[2] << endl;
    cout << " Fourth element of B[3] = " << B[3] << endl;
    cout << " Fifth element of  B[4] = " << B[4] << endl;
    return 0;
}
```

لاحظ: يتم حساب الفراغات عند الاخال كمدخل وكذلك حساب حجم الذاكرة بشكل دقيق.

RUN

```
enter string for A
abcdefghijklmnopqrstuvwxyz
 All elements of   A    = abcde
 First element of  A[0] = a
 Second element of A[1] = b
 Third element of  A[2] = c
 Fourth element of A[3] = d
 Fifth element of  A[4] = e
 All elements of   B    = abcde
 First element of  B[0] = a
 Second element of B[1] = b
 Third element of  B[2] = c
 Fourth element of B[3] = d
 Fifth element of  B[4] = e
```

لاحظ: المستخدم قام بإدخال جميع الحروف لكن عدد الحروف التي ظهرت وتمت معالجتها في البرنامج فقط هي بعدد المواقع الذاكرة التي تم الحجز والإعلان عنها مسبقاً وحسب اسبقية الادخال.

1. دالة (strcpy) تنسيب أو اسناد أو نسخ قيمة الى مصفوفة خيوط رمزية

لاحظ: تم استخدام مصفوفة فارغة المحتوى بنفس حجم المصفوفتان المراد تبديل محتواهم فيما بينهم

مثال(10): اكتب برنامج يقوم بتبديل محتوى خيطين رمزيين في ما بينهما.

```
1  #include <iostream>
2  #include <cstring>
3  using namespace std;
4  int main()
5  {
6      char X[20] , Y[20] , Z[20];
7      strcpy(X , "Good Morning");
8      strcpy(Y , "Good Afternoon");
9      cout<< "X first time = " << X << endl ;
10     cout<< "Y first time = " << Y << endl ;
11     cout<< "Z first time = " << Z << endl ;
12     strcpy(Z , X);
13     strcpy(X , Y);
14     strcpy(Y , Z);
15     cout<< "X second time = " << X << endl ;
16     cout<< "Y second time = " << Y << endl ;
17     cout<< "Z second time = " << Z << endl ;
18     return 0;
19 }
20
```

RUN

```
X first time = Good Morning
Y first time = Good Afternoon
Z first time =
X second time = Good Afternoon
Y second time = Good Morning
Z second time = Good Morning
```

لاحظ: ظهر محتوى المصفوفة Z فارغة (او بشكل رموز ربش في بعض الأحيان) لأننا لم نسند اليها قيمة في البداية وفيما بعد اخذت وبقت محتفظة بمحتوى المصفوفة X.

معالجة المصفوفات الرمزية غير المنتهية (Arrays Null-Terminated Strings)

2. دالة (strcmp( )) مقارنة الخيوط الرمزية.

- ✓ غير ممكن في لغة C أو C++

```
if (str1 < str2)
```

- ✓ حيث نستخدم الدالة strcmp() الموجودة في ملف الصديره <cstring> أو <string.h> .

```
strcmp(string1 , string2)
```

- ✓ الدالة strcmp() تعيد قيمة عددية صحيحه تكون:
  - اصغر من صفر عندما الخيط الرمزي الأول اصغر من الخيط الرمزي الثاني.
  - صفر عندما تشابه الخيطين.
  - اكبر من صفر عندما الخيط الرمزي الأول اكبر من الخيط الرمزي الثاني (وفقا لترتيب للقاموس).

```
#include <cstring>
int strcmp(const char *str1, const char *str2);
```

- ✓ The strcmp( ) function lexicographically compares two strings and returns an integer based on the outcome as shown here:

Value	Meaning
Less than zero	str1 is less than str2.
Zero	str1 is equal to str2.
Greater than zero	str1 is greater than str2.

مثـــال(11): اكتـــب برنـــامج لقـــراءة خيطـــا رمزيـــا وطباعـــة جملـــة "first string smaller than second string" أو "first string bigger than second string" أو string first string equal second string اعتمادا على كون الخيط الرمزي الأول اصغر أو أكبر أو يساوى الخيط الرمزى الثانى وعلى الترتيب؟

```
#include <iostream>
#include <cstring>
using namespace std;
int main()
{
    char X[10] , Y[10];
    cout << "enter first string" << endl;
    cin >> X;
    cout << "enter second string" << endl;
    cin >> Y;
    if (strcmp(X,Y)< 0)
        {
       // cout << strcmp(X,Y)<< endl;
        cout << "first string smaller than second string" << endl;
        }
    else if (strcmp(X,Y)> 0)
        {
       // cout << strcmp(X,Y)<< endl;
        cout << "first string bigger than second string" << endl;
        }
    else
        {
      //  cout << strcmp(X,Y)<< endl;
        cout << "first string equal second string" << endl;
        }
    return 0;
}
```

RUN

```
enter first string
abc
enter second string
m
first string smaller than second string
```

لاحظ: عند تفعيل هذا التعليمات سوف تعيد لنا الدالـة قيمـة امـا (-1) أو (1) أو (0) وحسـب ابجديـة ترتيـب الحروف الأولى للمدخلات.

2. دالة (strcmp( )) مقارنة الخيوط الرمزية.

مثال (12): **اكتب برنامج يحاكي حزوره يتم تثبيت حرف داخل البرنامج (مثل b) ثم يقوم بإظهار رسالة لإدخال حرف للمقارنة فاذا كان متساوي يظهر رسالة صح واذا مخالف يظهر رسالة خطأ؟**

```
#include <iostream>
#include <string.h>
using namespace std;
int main()
{
    char X[10];
    cout << "enter your letter" << endl;
    cin >> X ;
    if (strcmp(X,"b")) // different strings
        {
        //cout << strcmp(X,"b")<< endl;
        cout << "your letter is wrong" << endl;
        }
    else
        {
        //cout << strcmp(X,"b")<< endl;
        cout << "your letter is correct" << endl;
        }
    return 0;
}
```

RUN

لاحظ: عند تفعيل هذا التعليمات سوف تعيد لنا الدالة:

- ✓ **قيمة** (1) **ورسالة** your letter is wrong **اذا كان الخيط الرمزي المدخل اكبر من الخيط الرمزي المثبت في البرنامج.**
- ✓ **قيمة** (-1) **ورسالة** your letter is wrong **اذا كان الخيط الرمزي المدخل اصغر من الخيط الرمزي المثبت في البرنامج.**
- ✓ **قيمة** (0) **ورسالة** your letter is correct **اذا كان الخيط الرمزي المدخل مساوي من الخيط الرمزي المثبت في البرنامج.**

3. دمج (الحاق) (strcat()) خيط رمزي مع خيط رمزي أخر

```
#include <cstring>
char *strcat(char *str1, const char *str2);
```

- ✓ The strcat( ) function concatenates a copy of str2 to str1 and terminates str1 with a null. The null terminator originally ending str1 is overwritten by the first character of str2.
- ✓ The string str2 is untouched by the operation.
- ✓ If the arrays overlap, the behavior of strcat( ) is undefined.
- ✓ The strcat( ) function returns str1.
- ✓ Remember, no bounds checking takes place, so it is the programmer's responsibility to ensure that str1 is large enough to hold both its original contents and also those of str2.

- ✓ حيث نستخدم الدالة الموجودة في ملف الصديره <cstring> أو <string.h> strcat()

strcat(string1 , string2)

- ✓ الدالة تلحق (تدمج) string2 الى نهاية string1 والنتيجة في الذي يجب ان يكون طوله مناسب لاستيعاب ناتج الدمج.
- ✓ تبقى قيمة string2 محتفظة بقيمتها الاصلية.

4. دالة (strlen( )) تعيد طول خيط رمزي

```
#include <cstring>
size_t strlen(const char *str);
```

- ✓ The strlen( ) function returns the length of the null-terminated string pointed to by str. The null terminator is not counted.

- ✓ حيث نستخدم الدالة الموجودة في ملف الصديره <cstring> أو <string.h> strlen()

strlen (string)

- ✓ تعيد قيمة صحيحة تمثل طول الخيط (عدد الرموز).

```
char X[10]; الحجم الكلي المحجوز
strcpy( X , "Ali" );
cout << strlin(X) ;
```

- ✓ سوف تعيد الحجم الفعلي وهو (3).

معالجة المصفوفات الرمزية غير المنتهية (Arrays Null-Terminated Strings)

```
1  #include <iostream>
2  #include <string.h>
3  using namespace std;
4  int main()
5  {
6      char X[10], Y[10];
7      strcpy(X , "Ali");
8      strcpy(Y , "Ahmed");
9      cout << strlen(X) << endl;
10     cout << strlen(Y) << endl;
11     strcat(X , Y);
12     cout << strlen(X) << endl;
13     cout << X << endl;
14 }
```

مثال(13): اكتب برنامج يقوم بقياس طول خطين رمزيين ومن ثم يقوم بدمج الخيطين في خيط واحد وقياس طوله؟

RUN

```
3
5
8
AliAhmed
```

لاحظ:

- ✓ الحجم الكلي لـ X المحجوز (10).
- ✓ الحجم الفعلي لـ X قبل الدمج (3) .
- ✓ الحجم الفعلي لـ X بعد الدمج (8).

معالجة المصفوفات الرمزية غير المنتهية (Arrays Null-Terminated Strings)

5. strchr( ) دالة ترجع المؤشر الى تواجد الأول للـــ char في الخيط الرمزي

```
#include <cstring>
char *strchr(const char *str, int ch);
```

- ✓ The strchr( ) function returns a pointer to the first occurrence of the low-order byte of ch in the string pointed to by str.
- ✓ Returns a pointer to the first occurrence of ch in str.
- ✓ If no match is found, a null pointer is returned.

- ✓ ترجع الدالة strchr () المؤشر إلى التواجد لأول للمتغير ch مسار البحث طولة واحد بايت نوعه char في السلسلة التي يشير إليها str ابتداء من اليسار الى اليمين.
- ✓ تحذف الدالة الرموز التي قبل الرمز مدار البحث.
- ✓ إذا لم يتم العثور على تطابق ، يتم إرجاع مؤشر فارغ.

مثال(14): اكتب برنامج يقوم بالبحث عن أول رمز + في سلسلة رموز رياضية ويعيد طباعة السلسة ابتداء من أول رمز + الى نهاية السلسة ؟

```
1  #include <iostream>
2  #include <cstring>
3  using namespace std;
4  int main()
5  {
6      char c = '+';
7      cout<<(strchr("-/+*%+^+()=", c))<<endl;
8      return 0;
9  }
```

RUN

```
+*%+^+()=
```

مثال(15): اكتب برنامج يقوم بالبحث عن رمز + في سلسلة رموز رياضية ويطبع موجود ام لا ؟

```
1  #include <iostream>
2  #include <cstring>
3  using namespace std;
4  int main()
5  {
6      char c = '+';
7      if(strchr("-/+*%^=()", c))
8         cout<<"The variable is in the string"<< endl;
9      else
10        cout<<"The variable is not in the string"<< endl;
11     return 0;
12 }
13
```

RUN

```
The variable is in the string
```

معالجة المصفوفات الرمزية غير المنتهية (Arrays Null-Terminated Strings)

6. دالة ( )strrchr ترجع المؤشر الى تواجد الاخير بايت في الخيط الرمزي

```
#include <cstring>
char *strrchr(const char *str, int ch);
```

- ✓ The strrchr( ) function returns a pointer to the last occurrence of the low-order byte of ch in the string pointed to by str.
- ✓ If no match is found, a null pointer is returned.

- ✓ ترجع الدالة strrchr () مؤشراً إلى آخر تواجد لبايت ch منخفض الترتيب في السلسلة التي يشير إليها str .
- ✓ إذا لم يتم العثور على تطابق ، يتم إرجاع مؤشر فارغ.

مثال(16): اكتب برنامج يقوم بالبحث عن آخر رمز + في سلسلة رموز رياضيةٍ ويعيد طباعة السلسة ابتداءً من آخر رمز + الى نهاية السلسة ؟

```
1  #include <iostream>
2  #include <cstring>
3  using namespace std;
4  int main()
5  {
6      char c = '+';
7      cout<<(strrchr("-/+*%+^+()=", c))<<endl;
8      return 0;
9  }
```

RUN

```
+()=
```

لاحظ تم طباعة رمز + الأخير وما بعدها من خيط رمزي

معالجة المصفوفات الرمزية غير المنتهية (Arrays Null-Terminated Strings)

7. دالة (strstr( ) ترجع المؤشر الى التواجد الأول للخيط الرمزي الاول عند تطابقه مع الخيط الرمزي الثاني

```
#include <cstring>
char *strstr (const char *str1, const char *str2);
```

The strstr( ) function returns a pointer to the first occurrence in the string pointed to by str1 of the string pointed to by str2.

Returns a pointer to the first occurrence of str2 in str1.

It returns a null pointer if no match is found.

ترجع الدالة strstr () مؤشراً إلى التواجد الأول في السلسلة المشار إليها بواسطة str1 من السلسلة التي يشير إليها str2 .

إرجاع مؤشر إلى أول ظهور لـ str2 في str1 على str1 .

تقوم بإرجاع مؤشر فارغ إذا لم يتم العثور على تطابق.

مثال(17): اكتب برنامج يقوم بالبحث عن الخيط الرمزي1 (Mo) في الخيط الرمزي2 (Good Morning) ويطبع رموز السلسلة2 ابتداء من رموز السلسلة1؟

```
1  #include <iostream>
2  #include <cstring>
3  using namespace std;
4  int main()
5  {
6      cout<< strstr("Good Morning" ,"Mo")<<endl;
7      return 0;
8  }
```

RUN

```
Morning
```

معالجة السلاسل (الخيوط) الرمزية (The String Class)


```cpp
#include <iostream> //A short string demonstrati
#include <string>
using namespace std;

int main()
{
  string str1("Alpha");
  string str2("Beta");
  string str3("Omega");
  string str4;
  cout << "str1= " << str1 << "\n";
  cout << "str2= " << str2 << "\n";
  cout << "str3= " << str3 << "\n";
  cout << "str4= " << str4 << "\n";

// How to assign a string?
  str4 = str1;
  cout << "str4= " << str4 << "\n";

// How to concatenate two strings?
  str4 = str1 + str2;
  cout << "str4= " << str4 << "\n";

// How to concatenate a string with a C-string?
  str4 = str1 + " to " + str3;
  cout << "str4= " << str4 << "\n";

// How to compare strings?
  if(str3 > str1) cout << "str3 > str1\n";
  if(str3 == str1+str2)
     cout << "str3 == str1+str2\n";

/* A string object can also be assigned
   a normal string. */
  str1 = "This is a null-terminated string.\n";
  cout << "str1= " << str1 << "\n";

/* How to create a string object using another
   string object? */
  string str5(str1);
  cout << "str5= welcome " << str5 << "\n";

// How to input a string?
  cout << "Enter a string: ";
  cin >> str5;
  cout << "str5= " << str5 << "\n";
  return 0;
}
```

```
str1= Alpha
str2= Beta
str3= Omega
str4=
str4= Alpha
str4= AlphaBeta
str4= Alpha to Omega
str3 > str1
str1= This is a null-terminated string.

str5= welcome This is a null-terminated string.

Enter a string: welcome/C++/
str5= welcome/C++/
```

معالجة المصفوفات الرمزية غير المنتهية (Arrays Null-Terminated Strings)

1. دالة (strcpy) تنسيب أو اسناد أو نسخ قيمة الى مصفوفة خيوط رمزية

```
char s1[80], s2[80], s3[80];
s1 = "Alpha"; // can't do
s2 = "Beta"; // can't do

s3 = s1 + s2; // error, not allowed
```

غير مسموح في لغة C++

```
#include <cstring>
strcpy(s1, "Alpha");
strcpy(s2, "Beta");
strcpy(s3, s1);
strcat(s3, s2);
```

مسموح في لغة C++

The string Class

string + string
string + C-string
C-string + string

String is The String Class
C-string is Arrays Null-Terminated

THANK YOU