الحل الأمثل :

```
char plainText[100] ;
char ch ;
int i= 0 ;

while ( (ch = getche()) != '\r')
     plainText[i++] = ch ;

cout << "\n\n";

for (int i=0 ; i<strlen(plainText)-1 ; i++)
     cout << plainText[i] ;
```

و هنا تكون المصفوفة ذات حجم مناسب ، ويتم القراءة حرف بحرف إلى أن يتم إدخال حرف **carriage return** والذي هو enter .

لاحظ أن الدالة strlen(array) تقوم بإرجاع طول المصفوفة . أيضا الدالة getche() معرفه داخل الهيدر #include <conio.h> لا تنسى إضافته والا سيشتكي منك المترجم .

بقيت هناك التخلص من المسافات من النص الأصلي ، وسوف نتكلم عنها هنا ولن نذكرها مره أخرى ، حيث أنك اذا أدخلت مثلا wajdy essam ، يجب أن نتخلص من المسافة والا سوف يتم تطبيق داله التشفير عليها ، طبعا يمكنك تركها ، ويمكنك إلغائها و هذا ما أفضله .

```
while ( (ch = getche()) != '\r')
{
     if ( ch == ' ')
          continue ;

     plainText[i++] = ch ;
}
```

لاحظ تم اختبار الحرف هل هو المسافة ، فإذا كان كذلك فيتم الرجوع إلى الحلقة مره أخرى .

حل أخر بدون استخدام الدالة getch() ، واستخدام الدالة cin.get(array,size,endchar):

```
char plainText[100] ;

cin.get(plainText,100,'$');

for (int i=0 ; i<strlen(plainText) ; i++)
     cout << plainText[i] ;
```

قد يبدوا الحل مناسبا ، ولكن في حال أردت قرائه متغير ما بعد هذه الجملة cin.get(plaintext,100,'$') فلن تستطيع ، والسبب يعود إلى أن الحرف $ سوف يحل مكان القيمة التي تريد قراءتها ، لكي تفهم أكثر ، **قم بتشغيل المثال القادم** :

PDF created with pdfFactory Pro trial version www.pdffactory.com