

Strings and Structure

الخيوط الرمزية و التراكيب

This string is actually a one-dimensional array of characters which is terminated by a **null** character '\0'. Thus a null-terminated string contains the characters that comprise the string followed by a **null**.

The following declaration and initialization create a string consisting of the word "Hello". To hold the null character at the end of the array, the size of the character array containing the string is one more than the number of characters in the word "Hello."

```
char greeting[6] = {'H', 'e', 'l', 'l', 'o', '\0'};
```

If you follow the rule of array initialization, then you can write the above statement as follows

```
char greeting[] = "Hello";
```

Actually, you do not place the null character at the end of a string constant. The C++ compiler automatically places the '\0' at the end of the string when it initializes the array. Let us try to print above-mentioned string –

```
#include <iostream>
Using namespace std;
int main () {
    char greeting[6] = {'H', 'e', 'l', 'l', 'o', '\0'};

    cout << "Greeting message: ";
    cout << greeting << endl;

    return 0;
}
```

Function & Purpose

من ابرز الدوال التي تطبق على الخيوط الرمزية والتي تتواجد ضمن المكتبة < string > :

- 1 **strcpy(st1,st2);** دالة لاستنساخ خيط رمزي معين من خيط آخر ، ويمكن تحديد عدد الرموز المراد استقطاعها وبالشكل التالي: **strncpy(st1,st2,n);** تمثل **n** عدد الرموز المستقطعة.
- 2 **strcmp(st1,st2);** دالة للمقارنة بين الخيوط الرمزية.
- 3 **strlen(st);** دالة لحساب طول الخيط الرمزي .
- 4 **strcat(st1,st2);** دالة لدمج خيطين رمزيين بحيث تكون النتيجة في الخيط الرمزي الاول. ويمكن تحديد عدد الرموز التي تستقطع من الخيط الرمزي الثاني وتدمج الخيط الاول بواسطة: **strncat(st1,st2,n);** .
- ** ابرز الدوال التي تطبق على الخيوط الرمزية والتي تتواجد ضمن المكتبة < ctype.h > :
- 1 **isalnum(ch);** دالة لمعرفة الرمز اذا كان عبارة عن حرف ابجدي أو رقم ، ترجع هذه الدالة قيمة صفرية اذا كان الرمز لا يساوي قيمة ابجدية ولا رقم.
- 2 **isalpha(ch);** تختبر الرمز اذا كان حرف ابجدي او لا.
- 3 **islower(ch);** تختبر الرمز فيما اذا كان صغير ترجع قيمة والا ترجع صفر.
- 4 **isupper(ch);** تختبر الرمز اذا كان كبير فأنها ترجع قيمة.
- 5 **isdigit(ch);** تختبره اذا كان رقم ترجع قيمة.
- 6 **strupr();** تحول الحرف من صغير الى كبير.
- 7 **strlwr();** تحول الحرف من كبير الى صغير.
- 8 **strinv();** تعكس الخيط الرمزي.

-9 **strchr (s1,ch) ;** تقوم هذه الدالة بايجاد اول حدوث للرمز **ch** في الخيط الرمزي **s1**. اي تطبع الخيط الرمزي ابتداءً من الرمز **ch** .
cout << strchr("computer",'p');
Output: puter

-10 **strstr (s1,s2) ;** تعيد هذه الدالة اول حدوث للخيط الرمزي **s2** الموجود داخل الخيط الرمزي **s1** ,اي تطبع الخيط الرمزي من بداية الخيط الجزئي .

cout << strstr ("computer" , "put");
Output: puter

Following example makes use of few of the above-mentioned functions –

```
#include <iostream>
#include <string>
Using namespace std;
int main () {
    char str1[10] = "Hello";
    char str2[10] = "World";
    char str3[10];
    int len ;

    strcpy( str3, str1);
    cout << " copy str1 into str3: " << str3 << endl;
    if (strcmp(str1,str2)==0)
        cout<< "True";
    else
        cout<< "False";
    strcat( str1, str2);
    cout << " concatenates str1 and str2:" << str1 << endl;

    len = strlen(str1);
    cout << " length of str1 after concatenation = " << len << endl;

    return 0;
}
```

When the above code is compiled and executed, it produces result something as follows –

```
strcpy( str3, str1) : Hello
False
strcat( str1, str2): HelloWorld
strlen(str1) : 10
```

Output Function :

printf ();

هذه الدالة من الدوال الأساسية جداً في لغة السي وتستخدم للطباعة :

لو اردنا ان نطبع الجملة Hi I am Ali نستخدم الدالة printf بالشكل التالي :

```
printf ("Hi I am Ali ");
```

gets() and puts() functions:

The gets() function reads string from user and puts() function prints the string. Both functions are defined in <stdio.h> header file.

Let's see a simple program to read and write string using gets() and puts() functions.

```
#include<iostream>
```

```
#include<stdio.h>
```

```
Using namespace std;
```

```
Int main() {
```

```
    char name[50];
```

```
    printf("Enter your name: ");
```

```
    gets(name);
```

```
//reads string from keyboard
```

```
    printf("Your name is: ");
```

```
    puts(name);
```

```
//displays string
```

```
    return 0;
```

```
}
```

Output:

```
Enter your name: Sara
```

```
Your name is:Sara
```

Example: Write program to convert string to uppercase:

```
#include<iostream>
```

```
#include<string>
```

```

#include<ctype.h>

Using namespace std;

int main( ) {
    char str[80];
    int i;
    strcpy(str , "this is a test");
    for(i=0;str[i];i++)
        str[i]=toupper(str[i]);
    cout<<str;
    return 0;
}

```

Example: Write function to decided if a given string is palindrom or not (EVE , ABBA ,MADAM ,.....):

```

#include<iostream>
#include<stdio.h>
#include<ctype.h>
#include<string>
Using namespace std;

int palindrom(char a[],int size)
{
    int i,j,Index=0;

    j=size-1;
    for (i=0;i<size;i++)
    {
        if (a[i]!=a[j])
            return Index=1;
        j--;
    }
}

int main ()
{
    int Len, pal;
    char word[30];

    cout<<"enter word:\n";
    gets(word);
    Len=word.length();
    pal=palindrom(word,Len);
    if (pal==1)
        cout<<"not_palindrom";
    else

```

```

        cout<<"is_palindrom";

    return 0;
}

```

EXAMPLE:write a function called “n_of_space” to calculate the number of space in the given string:

```

#include<iostream>
#include<string>
Using namespace std;

void no_of_space(char s[])
{
    int i, num=0;

    for(i=0;i < strlen(s) ;i++)
        if (s[i]==' ')
            num++;
    cout<<"string :"<< s <<endl;
    cout<<"num="<< num;
}

int main()
{
    Char s[20]="number of space\0";

    no_of_space(s);
}

```

H.W :

- 1- write a function called “remove-space” to remove the spaces from the given string?
- 2- write a function “remove-repet” to remove all repetition of characters from a string?
- 3- write program to read string and calculate the number of alphabetic characters and the integer numbers in this string ,then give the length for this string?

Structures

- A **Structure** is a collection of related data items, possibly of different types.
- A **structure** type in C++ is called **struct**.
- A **structure** is **heterogeneous** in that it can be composed of data of different types.
- In contrast, **array** is **homogeneous** since it can contain only data of the same type.
- Structures hold data that belong **together**.

•Examples:

- o Student record: student id, name, major, gender, start year, ...
- o Bank account : account number, name, currency, balance, ...
- o Address book : name, address, telephone number, ...

Definition of a structure:

```
Struct <struct-name> {  
<type> <identifier_list> ;  
<type> <identifier_list> ;  
.....  
};
```

Example: Write structure has 3 Fields (day ,month ,year):

```
Struct Date {  
int day;  
int month;  
int year;  
};
```

مثال:

اكتب برنامج لإدخال بيانات الطالب (student structure) وهي الرقم الجامعي والاسم والعنوان والقسم والمرحلة ثم طباعتها باستخدام التركيب structure .

```
#include<iostream>  
Using namespace std;
```

```

int main ()
{

struct student
{
    int    no;
    char  name[30];
    char  address[20];
    char  department[15];
    int   level;
};
student st;

cout<<"enter the student number:\n";

cin>>st.no;
cout<<"enter the student name:\n";
cin>>st.name;
cout<<"enter the student address:\n";
cin>>st.address;
cout<<"enter the student department:\n";
cin>>st.department;
cout<<"enter the student level:\n";
cin>>st.level;

//printing the structure
cout<<"=====\n";
cout<<"student number ="<<st.no<<"\n";
cout<<"student name ="<<st.name<<"\n";
cout<<"student address ="<<st.address<<"\n";
cout<<"student department ="<<st. department <<"\n";
cout<<"student level ="<<st.level<<"\n";
cout<<"=====\n";
return 0;
}

```


Structures Arrays:

تعتبر المصفوفة من نوع structure من أهم التطبيقات على القیود حيث يعلن عن structure أولاً ثم يعلن عن متغير المصفوفة من نوع struct.

مثال:

اكتب برنامج لقراءة مصفوفة مكونة من 5 قیود حيث يشمل القید الواحد الحقول التالية (الاسم، المدينة، الشارع).

```
#include<iostream>
Using namespace std;

int main ()
{
    struct address
    {
        char name[20];
        char city[15];
        char street[20];
    };
    address info[5];
    int i;

    cout<<"Enter Information\n";
    for(i=0;i<5;i++)
    {
        cin>>info[i].name;
        cin>>info[i].city;
        cin>>info[i].street;
    }
    cout<<"**** Print Information ****\n";
    cout<<"=====\n";
    cout<<"No\tName\tCity\tStreet\n";
    for(i=0;i<5;i++)
    {
        cout<<i+1<<"\t";
        cout<<info[i].name<<"\t";
        cout<<info[i].city<<"\t";
        cout<<info[i].street<<"\t";
        cout<<"\n";
    }
    return 0;
}
```

}

H.W:

Write a program to read **N** structures, each structure has the fields (name ,number, city, and university) , calculate the number of student their cities is Basrah.