كلية العلوم

القسم : **الرياضيات**

السنة : **الثانية**

المادة : لغات البرمجة٢

المحاضرة : السابعة/عملي/

{{ مكتبة A to Z }}

Facebook Group : مكتبة A to Z

2026

كلية العلوم ، كلية الصيدلة ، الهندسة التقنية

يمكنكم طلب المحاضرات برسالة نصية (SMS) أو عبر(What's app-Telegram)على الرقم 0931497960

الجمهورية العربية السورية

جامعة طرطوس

كلية العلوم قسم الرياضيات

السنة الثانية

المادة: لغات برمجة2 _ عملي

المحاضرة السابعة

السجلات

• السجلات structure

بنية تجمع عدة بنود بيانات من أنواع مختلفة.

الصيغة العامة للسجل:

```
Struct  structName
{
بنود بيانات السجل من أنواع
 مختلفة
};
```

حيث:

Struct: الكلمة المفتاحية التي يجب أن يبدأ بها السجل

structName: اسم السجل

مثال: ليكن لدينا سجل طالب يتضمن (اسم الطالب name _ عمر الطالب age_ رقم الطالب Id _ عنوان الطالب add)

```
Struct  student
{
String name;
Int age ;
Int id;
String add;
};
```

نلاحظ أن السجل يضمن عدة بيانات مختلفة النوع

اسم الطالب وهو سلسلة محرفية string مثال ahmad

عمر الطالب وهو عدد صحيح int مثال 21

رقم الطالب وهو عدد صحيح int مثال 32

عنوان الطالب وهو سلسلة محرفية مثال Tartous

أدخال البيانات إلى السجل:

لإدخال بيانات إلى السجل يجب في التابع الرئيسي تعريف كائنات من هذا السجل تستطيع الوصول إلى بيانات السجل عن طريق (.)

يتم تعريف الكائن بالشكل: structName cp; حيث cp اسم الكائن.

مثال أدخال بيانات طالب إلى السجل السابق

```
#include<iostream>
#include<string>
Struct student
{
String name;
Int age ;
Int id;
String add;
};
Main()
{ student s1;
Cin>> s1.name >>s1.age>>s1.id>>s1.add;
}
```

تعريف كائن s1 من السجل

مثال2: أحد بيانات السجل مصفوفة

قم بإنشاء سجل طالب يتضمن (اسم الطالب name _ الرقم الجامعي id_ علامة خمس مواد)

نلاحظ أن علامات المواد هي مصفوفة من 5 عناصر

```
#include<iostream.h>
#include<string>
Struct student
{
String name;
Int age ;
Int id;
 Int mark [5];
};
```

```
Main ()
{student s1;
Cin>> s1.name >>s1.age>>s1.id;
For (int i=0; i<5; i++)
{Cin>>s1.mark[i];}
}
```

مثال3: عدد السجلات مصفوفة

قم بإنشاء سجل 10 طالب يتضمن (اسم الطالب name _ الرقم الجامعي id_ معدل الطالب avg_ عمر الطالب age)

نلاحظ أن عدد السجلات مصفوفة حدها الأعظمي 10

```
#include<iostream.h>
#include<string>
Const int size=10;
Struct student
{
String name;
Int age;
Int id;
 Float avg ;
};
```

```
Main ()
{student s[size];
Int n ;
Cin>>n;
For (int i=0; i<n; i++)
{Cin>> s[i].name >>s[i].age>>s[i].id>>s[i].avg;
}
}
```

مثال4:

اكتب برنامج لأنشاء سجل من عشرة مرضى لكل مريض المعلومات التالية (رقم المريض id _ اسم المريض name_ العمر age_ رقم الغرفة n_room _عدد أيام الرقود day) ويستدعي تابع لأدخال بيانات السجل وتابع لطباعة معلومات المرضى الذين مر على رقودهم 3 أيام؟

نلاحظ أن كائنات السجل هي مصفوفة حدها الأعظمي n

نعرف أولا السجل ثم التوابع ثم التابع الرئيسي

```
#include<iostream.h>
#include<string>
Const int size=10;
Struct info
{
String name;
Int age;
Int id;
Int n_room;
Int day;
};
Void read ( info e[size]  , int n)
{ For (int i=0; i<n; i++)
{ cin>>e[i].name>>e[i].age>>e[i].id>>e[i].n_room>>e[i].day;}
}
Void process ( info e[size]  , int n)
{ For (int i=0; i<n; i++)
{    If(e[i].day>3)
{ cout<< e[i].name<<  "    "<<e[i].age<<endl;
Cout<< e[i].id<<"    " << e[i].n_room<<endl;
Cout<< e[i].day;
}}}
Main()
{ info e[size];
Int n;
Cin>>n;
Read( e,n);
Process(e,n);  }
```

مثال 5: اكتب برنامج للتعامل مع الأعداد العقدية والمطلوب:

1. **إجرائية read لقراءة عدد عقدي.**
2. **إجرائية print لطباعة عدد عقدي.**
3. **تابع(دالة) sum لحساب مجموع عددين عقديين.**
4. **تابع (دالة) mult لحساب جداء عددين عقديين.**

نلاحظ أن العدد العقدي يتألف من قسم حقيقي وقسم تخيلي من الشكل a+ bi لذلك سوف نقوم بأنشاء سجل للعدد العقدي.

```
#include<iostream.h>
Struct num
{
Int re;
Int img;
};
Void read (num &x)
{cin>> x.re>>x.img;}
Void print (num &x)
{cout<<x.re<<"+"<<x.img<<"i";}
num sum ( num&x, num &y)
{ num  x1;
X1.re= x.re + y.re;
X1.img= x.img + y.img ;
Return x1;
}
num mult ( num&x , num&y)
{
num   x1;
X1.re=(x.re*y.re) - (x.img* y.img)  ;
X1.img=(x.re* y.img) + ( y.re*re.img);
Return x1;
}
```

```
Main()
{num x,y,z ;
read(x);
read(y);
print (x);
cout<<"\n";
print(y);
cout<<"\n";
z= sum(x,y);
print(z);
cout<<"\n";
z=mult(x,y);
cout<<"\n";
}
```

ملاحظة: جمع عددين عقديين هو عدد عقدي

القسم الحقيقي له = مجموع القسمين الحقيقيين للعددين.

القسم التخيلي له = مجموع القسمين التخيليين للعددين.

ضرب عددين عقديين نستخدم خاصية النشر

$$(a + bi) * (c + di) = ac + adi + cbi + bdi^2$$
$$= (ac - bd) + (ad + cb)i$$

قسم تخيلي — قسم حقيقي

مثال 6:

قم بإنشاء سجل 10 طالب يتضمن (اسم الطالب name _ الرقم الجامعي id_ علامة خمس مواد والمعدل avg) والمطلوب:

-ادخال بيانات الطلاب.

-ترتيب سجلات الطلاب حسب المعدل تنازليا.

-طباعة اسم الطالب صاحب اعلى معدل.

```
#include<iostream.h>
#include<string>
Const int size=10;
Struct student
{
String name;
Int id;
 Float avg ;
Int mark[5];
};
```

بما أن الطالب يدخل علاماته أي انه ليس المسؤول عن ادخال المعدل بل البرنامج يحسبه

```
Main ()
{student s[size];
Int n ;
Cin>>n;
Int sum=0;
Float avrg;
For (int i=0; i<n; i++)
{Cin>> s[i].name >>s[i].id;
For (int j=0; j<5; j++)
{Cin>>s[i].mark[j];
      Sum=sum+s[i]. mark[j];}
avrg=sum/5;
S[i].avg=avrg;
Sum=0; }
Int L;
for (int i=0; i<n; i++)
{for (int j=0; j<n; j++)
{if(s[j]. avg<s[j+1]. avg)
{L=s[j];
 s[j]=s[j+1];
s[j+1]=L;}
}}
Int m;
Int max=s[0].avg;
for (int i=0; i<n; i++)
{ if(s[i]. avg<max)
{max=s[i].avg;
m=i;}}
Cout<<s[m].name;
}
```

إعادة تصفير المجموع من اجل الطالب التالي

الترتيب حسب المعدل تنازليا

رقم كائن السجل صاحب أعلى معدل

طباعة اسم الطالب من السجل ذو الرقم m

السجلات المتداخلة:

هي عبارة عن سجلات تحتوي داخلها سجلات أخرى أي أن السجل الرئيسي يحتوي على مفردات وكل مفردة هي سجل بحد ذاتها (سجل فرعي).

مثال 7:

قم بإنشاء سجل 5 طالب يتضمن (اسم الطالب name (الاسم الاول –الاسم الثاني) _ الرقم الجامعي id_ والمعدل avg – العنوان (المدينة والشارع) – العمر(سنة وشهرويوم)) والمطلوب:

-تابع ادخال بيانات الطلاب.

-تابع طباعة أسماء الطلاب الذين ولدوا في شهر 3.

```
#include<iostream.h>
#include<string>
Const int size=10;
Struct names
{ string f-name;
String L-name;};
Struct address
{ string city;
String street;};
Struct birthday
{ int year;
Int mounth;
Int day;};
Struct student
{
names name;
address add;
birthday birth;
Int id;
Float avg;  };
```

سجل فرعي الاسم

سجل فرعي العنوان

سجل فرعي العمر

السجل الرئيسي student

```
Void read (student s[size]  , int n)
{ For (int i=0; i<n; i++)
{ cin>>s[i].name.f-name>> s[i].name.L-name;
Cin>>s[i].birth.year>> s[i].birth.mounth >> s[i].birth.day;
Cin>> s[i].add.city>> s[i].add.street;
Cin>> s[i].id>>s[i].avg;}
}
Void process ( student s[size]  , int n)
{ For (int i=0; i<n; i++)
{    If(s[i].birth.mounth==3)
{ cout<< s[i].name.f-name<< s[i].name.L-name;
}}}
Main()
{ student s[size]  ;
Int n;
Cin>>n;
Read( s,n);
Process(s,n);  }
```