

مقدمات نوشتن فایل

include <fstream>

ifstream

فراخوانی برای خواندن از فایل

ofstream

فراخوانی برای نوشتن از فایل

fstream

فراخوانی برای خواندن و نوشتن از فایل

۱۰ عدد از کاربر گرفته و هر یک از آنها را در یک خط از فایل ذخیره می کند.

```
# include <fstream>
main()
{
int adad;
ofstream fout ("file.txt");
cout<<"enter 10 number"<<endl;
for (int i=1; i<=10; i++)
{
cin>>adad;
fout<< adad<<endl
}
fout.close();
getch()
}
```

ذخیره یک جمله در فایل با فشردن کارکتر «.»

```
int adad;  
char ch;  
ofstream fout("file.txt");  
cout << "enter a sentence:" << endl;  
while (ch != '.')  
{  
    ch=getch();  
    cout << ch;  
    fout << ch;  
}  
fout.close;
```

خواندن ده عدد از یک فایل و پیدا کردن بزرگترین عدد

```
int adad, max;
ifstream fin("file.txt");
cout<<"number are:"<<endl;
for(int i=1; i<=10;i++)
{
fin>>adad;
cout<<adad<<endl;
if (i==1)
max=adad;
if (adad>max)
max=adad;
}
cout<<"max is:"<<max;
getch();
```

یک دفتر تلفن

برنامه زیر، چند نام و تلفن مربوط به هر یک را به ترتیب از کاربر دریافت کرده و در فایل به نام **PHONE.TXT** ذخیره می کند. کاربر برای پایان دادن به ورودی باید عدد 0 را تایپ کند.

```
#include <fstream>
#include <iostream>
using namespace std;
int main()
{ ofstream phonefile("PHONE.TXT");
  long number;
  string name;
  cout << "Enter a number for each name. (0 for quit): ";
  for ( ; ; )
  { cout << "Number: ";
    cin >> number;
    if (number == 0) break;
    phonefile << number << ' ';
    cout << "Name: ";
    cin >> name;
    phonefile << name << ' ';
    cout << endl;
  }
}
```

جستجوی یک شماره در دفتر تلفن

این برنامه، فایل تولید شده توسط برنامه قبل را به کار می‌گیرد و درون آن به دنبال یک شماره تلفن می‌گردد:

```
int main()
{ ifstream phonefile("PHONE.TXT");
  long number;
  string name, searchname;
  bool found=false;
  cout << "Enter a name for findind it's phone number: ";
  cin >> searchname;
  cout << endl;
  while (phonefile >> number)
  { phonefile >> name;
    if (searchname == name)
    { cout << name << ' ' << number << endl;
      found = true;
    }
    if (!found)
  cout << searchname << " is not in this phonebook." << endl;
}
```

ساختارها

ساختارها شبیه آرایه‌ها بوده بدین صورت که یک نوع داده گروهی است که فضای پیوسته از حافظه اصلی را اشغال می‌نماید. اما عناصر ساختار الزاماً از یک نوع نمی‌باشند بلکه اعضای یک ساختار می‌توانند از نوع‌های مختلفه از قبیل `char` ، `int` ، `float` ، ... باشند.

تعريف ساختار

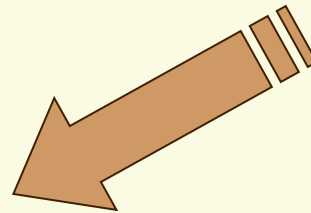
نام ساختار



```
struct time
```

```
{  
int hour ; // 0 – 23  
int minute ; // 0 – 59  
int second; //  
};
```

اعضا ساختار




```
struct account {  
int acc_no ;  
char acc_type;  
char name[80] ;  
float balance ;  
};
```

- ◆ ساختار account دارای چهار عضو می باشد.
- ◆ acc_no شماره حساب از نوع int
- ◆ acc_type نوع حساب از نوع char
- ◆ name مشخصات صاحب حساب از نوع رشته 80 کارکتری
- ◆ balance مانده حساب از نوع float

◆ به دو صورت می توان اعلان یک متغیر از نوع ساختار را نمایش داد :

```
struct account {  
int acc_no;  
char acc_type;  
char name[80];  
float balance;  
} cust1, cust2, cust3;
```

روش اول :

```
struct account {  
int acc_no ;  
char acc_type;  
char name[80];  
float balance;  
};  
account cust1, cust2, cust3;
```

روش دوم :

به ساختارها می توان مقدار اولیه نیز تخصیص داد

```
account cust = {4236, 'r', "Nader Naderi" , 7252.5};
```

دست‌رسی به عناصر یک ساختار

بمنظور دسترسی به عناصر یک ساختار از عملگر . استفاده می‌گردد . عملگر . جزء عملگرهای یکتائی می‌باشد.

مثال:

cust .acc_no = 4236

cust .acc_type = 'r'

cust . name = "Nader Naderi"

cust . balance = 7252.5



عضو یک ساختار خود می تواند یک ساختار دیگر باشد.

```
struct date {  
int month;  
int day;  
int year;  
};  
struct account {  
int acc_no ;  
char acc_typer;  
char name[80];  
float balance ;  
date lastpay ; };  
  
account x, y ;  
  
x.lastpay.month  
x.lastpay.year
```

برنامه ای بنویسید که نام و نام خانوادگی و معدل ۱۰ دانشجو را از ورودی بخواند و مشخصات دانشجوی نمونه را چاپ کند.

```
struct student {
    char name [31];
    char family[31];
    float avg;
};

void main(void)
{
    struct student s, max;
    int i;
    cin >> s.name >> s.family >> s.avg;
    max=s;
    for (i=1; i<10; i++)
    {
        cin >> s.name >> s.family >> s.avg;
        if (max.avg < s.avg)
            max = s;
    }
    cout << "name =" << max.name << endl ;
    cout << "family =" << max.family << endl;
    cout << "avg   =" << max.avg << endl;
    getch();
}
```


برنامه ای بنویسید که مشخصات ۱۰ دانشجو شامل نام و نام خانوادگی و معدل را از ورودی خوانده و آنها را بر اساس معدل مرتب کرده و چاپ کند.

```
struct student {
    char name [31];
    char family [31];
    float avg;
};

void main (void)
{
    int i;
    struct student s[10], temp;
    for(i=0; i<10; i++)
        cin >> s[i].name >> s[i].family >> s[i].avg;

    for (i=1; i<10; i++)
        for (j=0; j <10-i; j++)
            if (s[j].avg > s[j+1].avg)
            {
                temp = s[j];
                s[j] = s[j+1];
                s[j+1] = temp;
            }
    for (i=0; i<10; i++)
        cout << s[i].name << s[i].family << s[i].avg << endl;
}
```