```
/*
 * Advanced programming
 * Alireza Akhavan Pour
 * Akhavan@AlirezaWeb.com
 * date: 1394/11/12
 */

int main()
{
	cout<< "In the name of God";
	Lecture_2();
	return 0;
}

string Lecture_2()
{
	cout << “1.Vectors" <<endl;
	cout << “2.call by reference & call by value" <<endl;
	cout << “3.struct" <<endl;
	return ":)"
}
```

بردارها - vector

```cpp
#include <iostream>
#include <vector>
#include <algorithm>
using namespace std;

int main()
{
    vector<double> ages;
    double age;
    while (cin >> age)
        ages.push_back(age);

    double sum = 0;
    for (int i = 0; i < ages.size(); ++i)
        sum += ages[i];

    cout << "Average is: " << sum/ages.size() << endl;
    sort(ages.begin(), ages.end());
    cout << "The youngest one has " << ages[0] << " years old" <<endl;
    cout << "The oldest one has " << ages[ages.size()-1] << " years old" << endl;
    cout << "Median is: " << ages[ages.size()/2] <<endl;
    system("pause");
}
```

برای استفاده از vector این سرایند نیاز است

برای استفاده از تابع sort این سرایند نیاز است

دنباله‌ای از عناصر double، اندازه‌ی این دنباله در ابتدا صفر است.

تا موقعی که جریان ورودی به انتها نرسیده

به انتهای دنباله‌ی ages یک عنصر اضافه کن و مقدار age را در آن قرار بده

طول بردار ages ؟ یا تعداد عناصر این دنباله چند تاست؟

session2\1_vector_1

session2\2_vector_2

```cpp
#include <vector>
#include <string>
#include <iostream>
using namespace std;

int main()
{
    vector<int> my_vector;
    // vec[0] = 10; =>Error, why?!
    my_vector.push_back(10);
    my_vector[0] = 12;

    vector<double> v(4);
    v[0] = 1.5; v[1] = 3.7;
    v[2] = 9.2; v[3] = 41;

    vector<string> names(3);
    names [0] = "Ali";
    names [1] = "Reza";
    names [2] = "Hasan";

    //names[2] = 100 => Error, why?!;

    vector<double> vd(50, 3.2);
    //vd[50] = 2.6; =>Error, Why?!
}
```

این بردار هنگام ایجاد خالی است و هیچ عنصری ندارد.

خطای زمان اجرا: این بردار هنوز عنصری ندارد و در نتیجه خانه‌ی شماره ۰ هم بی معنیست.

بعد از **push_back**، یک خانه به بردار اضافه شد، حال دسترسی به خانه‌ی ۰ معنی دارد.
این دستور مقدار ۱۲ را جایگزین مقدار ۱۰ خانه‌ی شماره ۰ این بردار می‌کند.

ایجاد برداری از نوع **double** با نام **v** و دارای ۴ خانه از شماره‌های ۰ تا ۳

ایجاد برداری از نوع **string** با نام **names** و دارای ۳ خانه از شماره‌های ۰ تا ۲

نوع عناصر بردار **names** رشته است، نه عدد!

ایجاد برداری از نوع **double** با نام **vd** و دارای ۵۰ خانه از شماره‌های ۰ تا ۴۹ و با مقدار اولیه‌ی ۳.۲

طول بردار ۵۰ است و تا خانه‌ی ۴۹ آن تعریف شده است؛ دسترسی به خانه‌ی ۵۰ خطای زمان اجرا می‌دهد

Vector of vectors

```
int main()
{
// An empty vector of vectors. The space
// between the 2 '>' signs is necessary
   vector<vector<int> > v2d;

// Now we'll try to create a 3 by 5 "matrix".
// First, create a vector with 5 elements
   vector<int> v2(5, 99);

// Now create a vector of 3 elements.
// Each element is a copy of v2
   vector<vector<int> > v2d2(3,v2);

// Print out the elements
   for(int i=0;i<v2d2.size(); i++) {
      for (int j=0;j<v2d2[i].size(); j++)
        cout << v2d2[i][j] << " ";
      cout << endl;
   }
   system("pause");
}
```

Call by value

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

void sum_with_five(int x)
{
    x +=5;
}

int main()
{
    int a = 20;
    sum_with_five(a);
    cout << a << endl;
    system("pause");
    return 0;
}
```

?

session2\3_functions_1_call_by_value

Call by reference

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

void f(int x, int& y)
{
    x = 5;
    y = 10;
}

int main()
{
    int a = 1;
    int b = 2;
    f(a, b);
    cout << a << endl;
    cout << b << endl;
    system("pause");
    return 0;
}
```

آرگومان دوم این تابع از نوع by reference یا با ارجاع رد می‌شود

? ?

session2\4_functions_2_call_by_reference

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

void g(int& y)
{
    y = 30;
}

void f(int x)
{
    g(x);
    cout << x << endl;
}

int main()
{
    int a = 10;
    f(a);
    cout << a << endl;

    cin.get();
    return 0;
}
```

چه اعدادی در خروجی چاپ می‌شود؟

session2\5_functions_nested

Struct

فرض کنید اطلاعات خودتان را میخواهید نگاهداری کنید!

```
1  std::string myName;
2  int myBirthYear;
3  int myBirthMonth;
4  int myBirthDay;
5  int myHeightInches;
6  int myWeightPounds;
```

فرض کنید با این اطلاعات زیاد کار دارید،
اطلاعات چند شخص هم میخواهیم نگاهداری کنیم؛
چه مشکلی حس میکنید؟!

aggregate data type
یا
Struct

تعریف و معرفی ساختار یا رکورد یا استراکچر:

```
1  struct Employee
2  {
3      short id;
4      int age;
5      double wage;
6  };
```

استفاده از یک ساختار:

```
1  Employee joe; // create an Em
2  ployee struct for Joe
   Employee frank; // create an
   Employee struct for Frank
```

دسترسی به اعضای یک ساختار (دسترسی به فیلدها):

```
Employee joe; // create an Employee struct for Joe
joe.id = 14; // assign a value to member id within
struct joe
joe.age = 32; // assign a value to member age withi
n struct joe
joe.wage = 24.15; // assign a value to member wage
within struct joe

Employee frank; // create an Employee struct for Fr
ank
frank.id = 15; // assign a value to member id withi
n struct frank
frank.age = 28; // assign a value to member age wit
hin struct frank
frank.wage = 18.27; // assign a value to member wag
e within struct frank
```

```
int totalAge = joe.age + frank.age;

if (joe.wage > frank.wage)
    cout << "Joe makes more than Frank\n";
else if (joe.wage < frank.wage)
    cout << "Joe makes less than Frank\n";
else
    cout << "Joe and Frank make the same amount\n";

// Frank got a promotion
frank.wage += 2.50;

// Today is Joe's birthday
++joe.age; // use pre-increment to increment Joe's age by 1
```

```
struct Employee
{
    short id;
    int age;
    double wage;
};

Employee joe = { 1, 32, 60000.0 };
// joe.id = 1, joe.age = 32, joe.wage = 60000.0

Employee frank = { 2, 28 };
// frank.id = 2, frank.age = 28, frank.wage = 0.0 (default initialization)
```

```cpp
#include <iostream>
using namespace std;
struct Employee
{
    short id;
    int age;
    double wage;
};

void printInformation(Employee employee)
{
    cout << "ID:   " << employee.id << endl;
    cout << "Age:  " << employee.age << endl;
    cout << "Wage: " << employee.wage << endl;
}

int main()
{
    Employee joe = { 14, 32, 24.15 };
    Employee frank = { 15, 28, 18.27 };

    // Print Joe's information
    printInformation(joe);

    cout << endl;

    // Print Frank's information
    printInformation(frank);
    system("pause");
    return 0;
}
```

struct