



FUTURE TECHNOLOGY
CÔNG NGHỆ TƯƠNG LAI

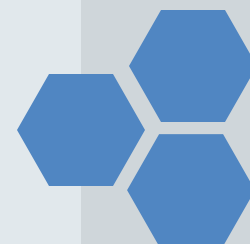
TRƯỜNG ĐẠI HỌC KINH TẾ KỸ THUẬT CÔNG NGHIỆP

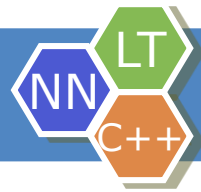
Khoa Công Nghệ Thông Tin



Phần 2. NN lập trình C++

CHƯƠNG 9 CẤU TRÚC (STRUCTURE)





Chương 9 – Cấu trúc (Structure)

1

Khái niệm cấu trúc

2

Khai báo cấu trúc

3

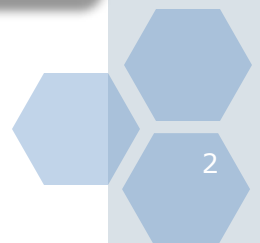
Định nghĩa cấu trúc bằng typedef

4

Thao tác trên biến cấu trúc

5

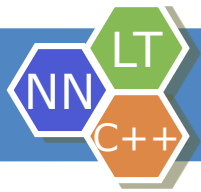
Truyền biến cấu trúc cho hàm



9.1 Khái niệm cấu trúc

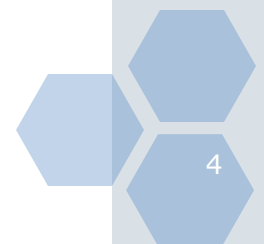
- ✦ Một cấu trúc bao gồm các mẫu dữ liệu, không nhất thiết cùng kiểu, được nhóm lại với nhau.
- ✦ Một cấu trúc có thể bao gồm nhiều mẫu dữ liệu như vậy.





9.1 Khái niệm cấu trúc

- ❑ Việc định nghĩa cấu trúc sẽ tạo ra kiểu dữ liệu mới cho phép người dùng sử dụng chúng để khai báo các biến kiểu cấu trúc .
- ❑ Các biến trong cấu trúc được gọi là các phần tử của cấu trúc hay thành phần của cấu trúc

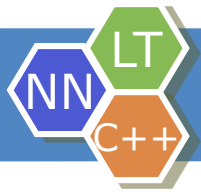


9.2 Khai báo cấu trúc

```
struct <tên cấu trúc>
{
    ...
    <kiểu>    <tên biến thành phần>;
    ...
};
```

Ví dụ:

```
struct book
{
    char    ten_sach[25];
    char    tac_gia[20];
    int     lan_xb;
    float   gia_thanh;
};
```



9.2 Khai báo cấu trúc

`struct` <tên cấu trúc> <danh sách biến cấu trúc>;

Ví dụ:

```
struct book books1;
```

Cách khác

```
struct book {  
    char    ten_sach [25];  
    char    tac_gia  [20];  
    int     lan_xb;  
    float   gia_thanh;  
} books1, books2;
```

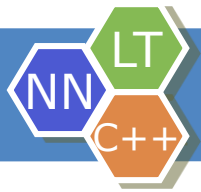
```
struct book books1, books2;  
hoặc  
struct book books1;  
struct book books2;
```

9.2 Khai báo cấu trúc

- ❑ Một cấu trúc có thể lồng trong một cấu trúc khác.
Tuy nhiên, một cấu trúc không thể lồng trong chính nó.

```
struct date
{
    int    day;
    int    month;
    int    year;
}

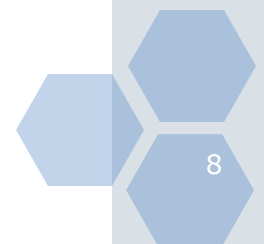
struct student
{
    int    no;
    char    name[20];
    struct date birthday;
    float    mark;
};
```



9.2 Khai báo cấu trúc

- ❑ Một áp dụng thường gặp là mảng cấu trúc
- ❑ Một kiểu cấu trúc phải được định nghĩa trước, sau đó một biến mảng có kiểu đó mới được khai báo
- ★ Ví dụ: **struct book books[50];**
- ❑ Để truy cập vào thành phần author của phần tử thứ năm của mảng books:

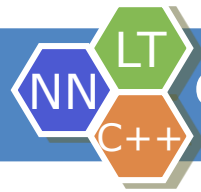
books[4].tac_gia



9.2 Khai báo cấu trúc

- ❑ Mảng cấu trúc được khởi tạo bằng cách liệt kê danh sách các giá trị phần tử của nó trong một cặp dấu móc
- ❑ Ví dụ:

```
struct unit
{
    char ch;
    int i;
};
struct unit series[3] =
    {{ 'a', 100 } { 'b', 200 } { 'c', 300 } };
```



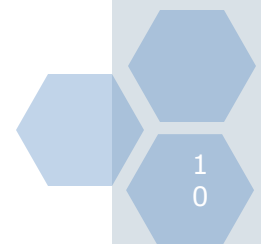
9.3 Định nghĩa cấu trúc bằng typedef

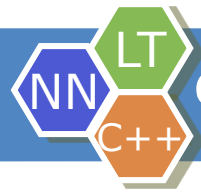
- ❑ Một kiểu dữ liệu có thể được định nghĩa bằng cách sử dụng từ khóa typedef
- ❑ Nó không tạo ra một kiểu dữ liệu mới, mà định nghĩa một tên mới cho một kiểu đã có.
- ❑ Cú pháp:

```
typedef <kiểu đã có> <tên mới>;
```

- ❑ Ví dụ:

```
typedef      int      so_nguyen;  
typedef      float    so_thuc;
```





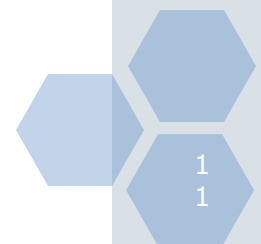
9.3 Định nghĩa cấu trúc bằng typedef

★ Ví dụ: typedef int* PTR_INT;

=> *chỉ cần viết:*

```
PTR_INT        px;
```

```
// tương đương với khai báo    int*    px;
```



9.3 Định nghĩa cấu trúc bằng typedef

```
typedef struct
{
    ...
    <kiểu> <tên biến thành phần>;
    ...
} <tên cấu trúc>;
```

Ví dụ: Định nghĩa kiểu phân số như sau:

```
typedef struct
{
    int tuso, mauso;
} PHANSO;
PHANSO ps1, ps2;
```



9.4 Thao tác trên biến cấu trúc

- Các phần tử của cấu trúc được truy cập thông qua việc sử dụng toán tử chấm (.) hoặc (->), toán tử này còn được gọi là toán tử thành viên - membership.

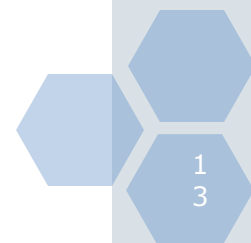
- Cú pháp:

`tên_biến_cấu_trúc.tên_thành_phần`

`tên_biến_cấu_trúc -> tên_thành_phần`

- Ví dụ:

```
cin>> books1.ten_sach;
```



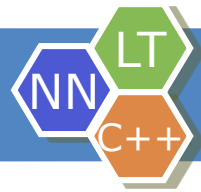
9.4 Thao tác trên biến cấu trúc

- Giống như các biến khác và mảng, các biến kiểu cấu trúc có thể được khởi tạo tại thời điểm khai báo

```
struct employee
{
    int    no;
    char   name[20];
};
```

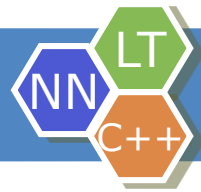
- Các biến emp1 và emp2 có kiểu employee có thể được khai báo và khởi tạo như sau:

```
struct employee emp1 = {346, "Abraham"};
struct employee emp2 = {347, "John"};
```



9.5 Truyền biến cấu trúc cho hàm

- ❑ Tham số của hàm có thể là một cấu trúc.
- ❑ Là một phương tiện hữu dụng khi muốn truyền một nhóm các thành phần dữ liệu có quan hệ logic với nhau thông qua một biến thay vì phải truyền từng thành phần một
- ❑ Kiểu của tham số thực phải trùng với kiểu của tham số hình thức.



9.5 Truyền biến cấu trúc cho hàm

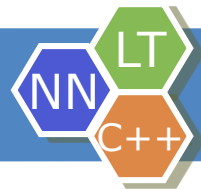
- ❑ Con trỏ cấu trúc được khai báo bằng cách đặt dấu * trước tên của biến cấu trúc.
- ❑ Toán tử -> được dùng để truy cập vào các phần tử của một cấu trúc sử dụng một con trỏ

★ Ví dụ:

```
struct book *ptr_bk;  
ptr_bk = &books;  
cout<<ptr_bk->author;
```

- ❑ Con trỏ cấu trúc được truyền vào hàm, cho phép hàm thay đổi trực tiếp các phần tử của cấu trúc.





9.5 Truyền biến cấu trúc cho hàm

- ❑ Khai báo và truy xuất biến cấu trúc khi sử dụng con trỏ:
- ❑ Với cấu trúc PHANSO ở trên
- ❑ Có khai báo:

```
PHANSO ps, *pps;
```

Truy xuất các biến thành phần của pps như sau:

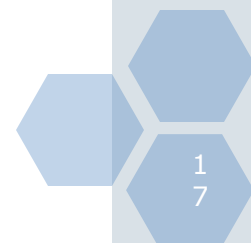
```
ps.tuso
```

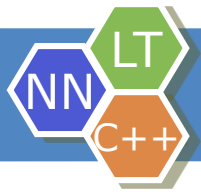
```
ps.mauso
```

Hoặc

```
pps ->tuso
```

```
pps ->mauso
```





VÍ DỤ STRUCT

Bài toán: Viết chương trình tạo một cấu trúc sản phẩm gồm các trường: mã sản phẩm, tên sản phẩm, số lượng, đơn giá, thành tiền.

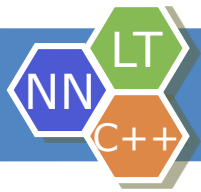
a) Nhập vào một danh sách n sản phẩm (không nhập trường thành tiền).

b) Viết hàm tính thành tiền cho các sản phẩm biết

$$\text{thành tiền} = \text{số lượng} * \text{đơn giá}$$

c) In ra màn hình danh sách sản phẩm vừa nhập dưới dạng bảng

d) In danh sách những sản phẩm có giá bán < 20.000 dưới dạng bảng.



VÍ DỤ STRUCT

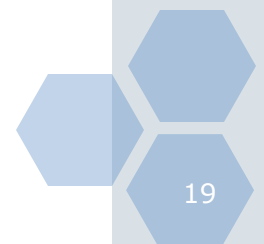
Khai báo thư viện cần dùng

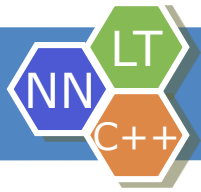
```
#include <iostream.h>
```

```
#include <iomanip.h>
```

Định nghĩa cấu trúc:

```
typedef struct
{
    char    masp[10];
    char    tensp[25];
    int     soluong;
    float   dongia;
    float   thanhtien
} san_pham;
```





VÍ DỤ STRUCT

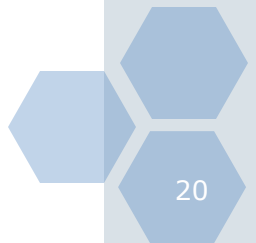
Khai báo hàm `main()` là hàm chính của chương trình.

```
int main()
```

Khai báo biến cấu trúc, biến điều khiển

```
san_pham    sp[50]; //sp là mảng kiểu cấu trúc
```

```
int n,i; //i là biến điều khiển của vòng for
```





VÍ DỤ STRUCT

Nhập giá trị cho số n

```
cout<<"Nhap so san pham "; cin>>n;
```

Sử dụng câu lệnh for nhập giá trị cho n sản phẩm

```
for (i=0; i<n; i++)
```

```
{  cin.ignore();
```

```
    cout<<"\n--Nhap san pham thu "<< i+1<<"--"<<endl;
```

```
    cout<<"Ma san pham:"; cin.getline(sp[i].masp,10);
```

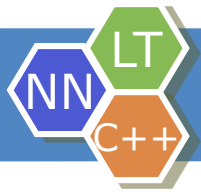
```
    cout<<"\nTen san pham:" ;
```

```
    cin.getline(sp[i].tensp,25);
```

```
    cout<<"\nSo luong san pham:";cin>>sp[i].soluong;
```

```
    cout<<"\nDon gia san pham:";cin>>sp[i].dongia;
```

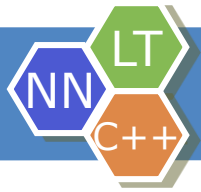
```
}
```



VÍ DỤ STRUCT

In DS sản phẩm vừa nhập ra màn hình:

```
cout<<"\n\n--- DANH SACH SAN PHAM ---"<<endl;
cout<<setw(10)<<"Ma san pham"<<setw(25)<<"Ten san pham";
cout<<setw(15)<<"So luong"<<setw(15)<<"Don gia";
cout<<setw(15)<<"Thanh tien"<<endl;
for (int i=0;i<n;i++)
{
    cout<<setw(15)<<sp[i].masp;
    cout<<setw(20)<<sp[i].tensp;
    cout<<setw(15)<<sp[i].soluong;
    cout<<setw(15)<<sp[i].dongia
    cout<<setw(15)<<sp[i].thanhtien<<endl;
}
```



VÍ DỤ STRUCT

Tính thành tiền cho các sản phẩm:

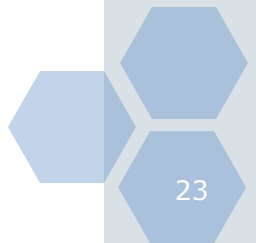
```
//tính thanh tien cho cac san pham
```

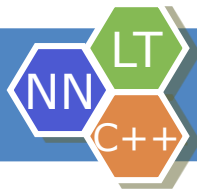
```
for (int i=0;i<n;i++)
```

```
{
```

```
    sp[i].thanhtien = sp[i].soluong*sp[i].dongia;
```

```
}
```

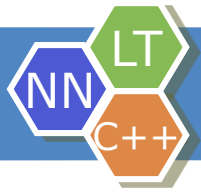




VÍ DỤ STRUCT

In DS sản phẩm có giá bán < 20000

```
cout<<"\n--Danh sach san pham co gia <20000--"<<endl;
cout<<setw(15)<<"Ma san pham"<<setw(20)<<"Ten san pham";
cout<<setw(15)<<"So luong"<<setw(15)<<"Don gia";
cout<<setw(15)<<"Thanh tien"<<endl;
for (i=0;i<n;i++)
    if (sp[i].dongia <= 20000)
    {
        cout<<setw(15)<<sp[i].masp;
        cout<<setw(30)<<sp[i].tensp;
        cout<<setw(15)<<sp[i].soluong;
        cout<<setw(15)<<sp[i].dongia;
        cout<<setw(15)<<sp[i].thanhtien<<endl;
    }
```

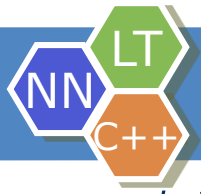



VÍ DỤ STRUCT

```
#include <iostream.h>

#include <iomanip.h>

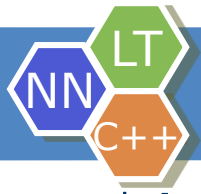
typedef struct
{
    char    masp[15];
    char    tensp[30];
    int     soluong;
    float   dongia;
} san_pham;
```



VÍ DỤ STRUCT

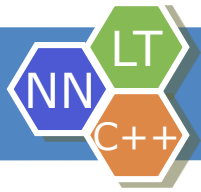
```
void nhapds (san_pham a[], int m)
{ for (int i=0; i<m; i++)
    { cin.ignore();
      cout<<"\n--Nhap san pham thu "<< i+1<<"--"<<endl;
      cout<<"Ma san pham:"; cin.getline(a[i].masp, 15);
      cout<<"\nTen san pham:"; cin.getline(a[i].tensp, 30);
      cout<<"\nSo luong san pham:"; cin>>a[i].soluong;
      cout<<"\nDon gia san pham:"; cin>>a[i].dongia;
    }
}

void  tinh_thanh_tien (san_pham a[], int m)
{ for (int i=0; i<m; i++)
    a[i].thanhtien = a[i].soluong*a[i].dongia;
}
```



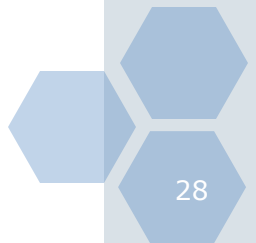
VÍ DỤ STRUCT

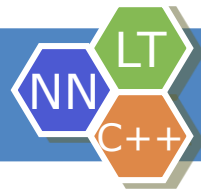
```
void inds(san_pham a[], int m)
{cout<<"\n\n--- DANH SACH SAN PHAM ---"<<endl;
cout<<setw(10)<<"Ma san pham"<<setw(25)<<"Ten san pham";
cout<<setw(15)<<"So luong"<<setw(15)<<"Don gia";
Cout<<setw(15)<<"Thanh tien"<<endl;
for (int i=0;i<m;i++)
{
    cout<<setw(10)<<a[i].masp;
    cout<<setw(25)<<a[i].tensp;
    cout<<setw(15)<<a[i].soluong;
    cout<<setw(15)<<a[i].dongia;
    cout<<setw(15)<<a[i].thanhtien<<endl;
}
}
```



VÍ DỤ STRUCT

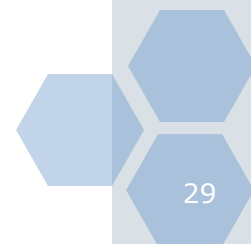
```
void spgiaduoai20000 (san_pham a[], int m)
{cout<<"\n--Danh sach san pham co gia <20000--"<<endl;
cout<<setw(15)<<"Ma san pham"<<setw(20)<<"Ten san pham";
cout<<setw(15)<<"So luong"<<setw(15)<<"Don gia";
Cout<<setw(15)<<"Thanh tien"<<endl;
for (i=0;i<n;i++)
    if (sp[i].dongia <= 20000)
    {
        cout<<setw(15)<<sp[i].masp;
        cout<<setw(30)<<sp[i].tensp;
        cout<<setw(15)<<sp[i].soluong;
        cout<<setw(15)<<sp[i].dongia;
        cout<<setw(15)<<sp[i].thanhtien<<endl;
    }
}
```

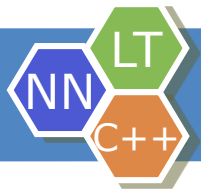




VÍ DỤ STRUCT

```
int main()
{ san_pham sp[50];
  int n;
  cout<<"Nhap so san pham: ";cin>>n;
  nhapds(sp, n);
  tinh_thanh_tien(sp, n);
  inds(sp, n);
  spgiaduoic20000(sp, n);
  return 0;
}
```





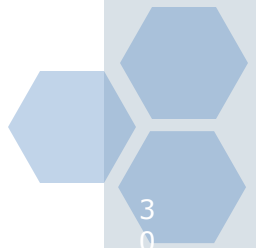
Câu hỏi củng cố bài

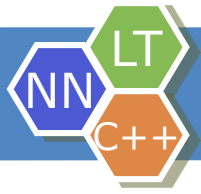
1. Khi sử dụng dữ liệu con trỏ kiểu cấu trúc, để sử dụng dữ liệu thành viên, ta sử dụng ký hiệu gì?

- A. Dấu .
- B. Dấu ->
- C. Dấu :
- D. Dấu ::



Multiple Choice



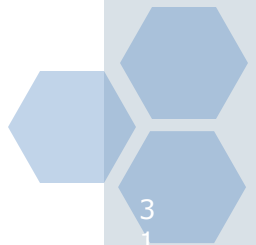


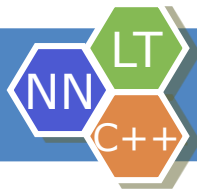
Câu hỏi củng cố bài

2. Khi sử dụng biến kiểu cấu trúc, để sử dụng dữ liệu thành viên, ta sử dụng ký hiệu gì?
- A. Dấu .
 - B. Dấu ->
 - C. Dấu :
 - D. Dấu ::



Multiple Choice





Câu hỏi củng cố bài

3. Cách khai báo biến cấu trúc trong C++?

- A. struct <Tên kiểu dl>{<Các dl thành phần>;
- B. struct <Tên kiểu dl>{<Các dl thành phần>}
[danh sách tên biến]
- C. struct <Tên kiểu dl>{<Các dl thành phần>}
[danh sách tên biến];
- D. struct <Tên kiểu dl>{<Các dl thành phần>}



Multiple Choice

4. Đoạn lệnh sau cho kết quả thể nào?

A. 1971

B. 1970

C. 1973

D. 1972

```
struct hocsinh {
    string hoten;
    int namsinh;
};

int main() {
    hocsinh hs[3];
    hocsinh *p = hs;
    for (int i = 0; i < 3; i++) {
        p->namsinh = 1991 + i;
        p++;
    }
    p = hs+1;
    p->namsinh -= 20;
    cout << hs[1].namsinh << endl;
    return 0;
}
```



Multiple Choice

5. Đoạn lệnh sau cho kết quả thể nào?

A. 1992

B. 1991

C. 1971

D. 1972

```
struct hocsinh {
    string hoten;
    int namsinh;
};

int main() {
    hocsinh hs[3];
    hocsinh *p = hs;
    for (int i = 0; i < 3; i++) {
        p->namsinh = 1991 + i;
        p++;
    }
    p = hs + 1;
    p->namsinh -= 20;
    cout << hs[0].namsinh << endl;
    return 0;
}
```



Multiple Choice

6. Đoạn lệnh sau cho kết quả thế nào?

A. 5, 5

B. 5, 25, 125

C. 5, 25

D. 5, 25, 625

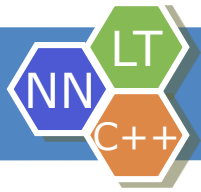
```
struct hocsinh {
    string hoten;
    int namsinh;
    double diem;
};

double bac3 (double a,int x) {
    double d = 1;
    for (int i = 0; i < x; i++) d *= a;
    return d;
}

int main() {
    hocsinh hs[3];
    hocsinh *p;
    p = hs;
    for (int i = 0; i < 3; i++) {
        p->diem = 5;
        p++;
    }
    cout << hs[0].diem << ", " << bac3(hs[1].diem,2);
```

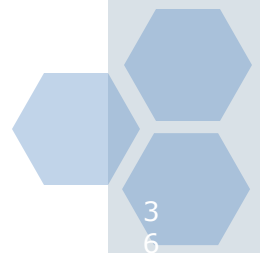


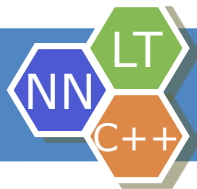
Multiple Choice



Câu hỏi lý thuyết

1. Nêu cách khai báo cấu trúc? Cho ví dụ.
2. Nêu cách định nghĩa cấu trúc? Cho ví dụ.





Bài tập

1. Để quản lý nhân viên của một cơ quan người ta lưu trữ các thông tin sau: Mã nhân viên, Tên nhân viên, Hệ số lương, Lương CB, Phụ cấp, Tổng lương.

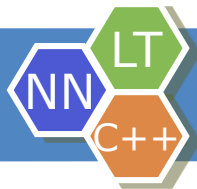
- a) Nhập vào một danh sách n nhân viên (không nhập trường Lương CB, Phụ cấp, Tổng lương).
- b) Viết hàm tính Lương CB, phụ cấp, tổng lương cho các nhân viên biết

$$\text{Lương CB} = \text{Hệ số lương} * 27500$$

$$\text{Phụ cấp} = 25\% * \text{Lương CB}$$

$$\text{Tổng lương} = \text{Lương CB} + \text{Phụ cấp}$$

- c) In ra màn hình danh sách nhân viên dưới dạng bảng
- d) Sắp xếp danh sách nhân viên theo thứ tự giảm dần của tổng lương. In ra danh sách sau khi sắp xếp.



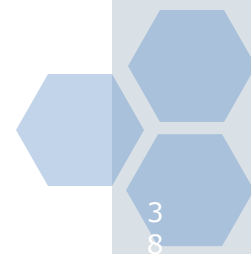
Bài tập

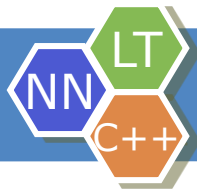
2. Để quản lý học sinh của một lớp người ta lưu trữ các thông tin sau: Mã học sinh, Tên học sinh, Giới tính, Điểm toán, Điểm lý, Điểm hóa, Điểm tb, Xếp loại.

a) Nhập vào một danh sách n học sinh (không nhập trường Điểm tb, Xếp loại).

b) Viết hàm tính Điểm tb cho các học sinh biết:

$$\text{Điểm tb} = (\text{Điểm toán} * 4 + \text{Điểm lý} * 3 + \text{Điểm hóa} * 2) / 9$$





Bài tập

c) Viết hàm điền xếp loại cho các học sinh biết

Nếu điểm tb ≥ 9 , Xếp loại Xuất sắc

Nếu $8 \leq \text{điểm tb} < 9$, Xếp loại Giỏi

Nếu $6.5 \leq \text{điểm tb} < 8$, Xếp loại Khá

Nếu $5 \leq \text{điểm tb} < 6.5$, Xếp loại Trung bình

Nếu điểm tb < 5 , Xếp loại Yếu

d) In ra màn hình danh sách học sinh với đầy đủ thông tin dưới dạng bảng

e) In ra danh sách các học sinh nữ xếp loại xuất sắc và loại giỏi.

