

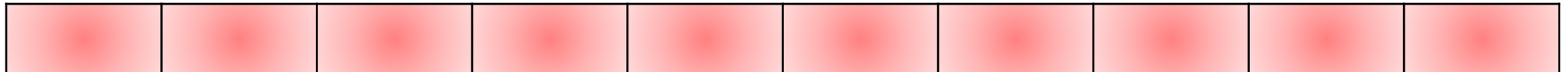
LÝ THUYẾT MẢNG MỘT CHIỀU

1. ThS. Nguyễn Hữu Lợi
2. ThS. Võ Duy Nguyên
3. TS. Nguyễn Tấn Trần Minh Khang

HÌNH ẢNH MẢNG MỘT CHIỀU

Hình ảnh mảng một chiều

— Hình vẽ minh họa mảng một chiều



— Mảng một chiều là một kiến trúc bao gồm nhiều ô được xếp liên tiếp nhau.

KHÁI NIỆM MẢNG MỘT CHIỀU

Khái niệm mảng một chiều

— Khái niệm: Mảng một chiều là một tập hợp các biến có cùng kiểu dữ liệu và cùng tên.

— Hình vẽ minh họa mảng một chiều



— Khái niệm: Mảng một chiều là một tập hợp các biến có cùng kiểu dữ liệu và cùng tên.

CHỈ SỐ

Chỉ số

- Qui ước: Các phần tử của mảng một chiều trong ngôn ngữ lập trình **C/C++** được đánh chỉ số (index) bắt buộc từ **0**.

0	1	2	3	4	5	6	7	8	9

- Như vậy các phần tử trong mảng một chiều có **n** phần tử được đánh chỉ số theo thứ tự từ **0** cho đến **$n - 1$** .

0	1	2	...	$i - 1$	i	$i + 1$...	$n - 2$	$n - 1$

KHAI BÁO MẢNG MỘT CHIỀU

Khai báo mảng một chiều

— Cú pháp:

```
KDL <TenBien>[SoPhanTuToiDa];
```

— Ví dụ 01:

```
+ int a[100];
```

+ Trong ví dụ trên ta nói **a** là mảng một chiều có tối đa 100 phần tử. Mỗi phần tử trong mảng có kiểu dữ liệu là kiểu số nguyên **int**.

Khai báo mảng một chiều

— Cú pháp:

KDL <TenBien>[SoPhanTuToiDa];

— Ví dụ 02:

+ `float b[20];`

+ Trong ví dụ trên ta nói **b** là mảng một chiều có tối đa 20 phần tử. Mỗi phần tử trong mảng có kiểu dữ liệu là kiểu số thực `float`.

Khai báo mảng một chiều

— Ví dụ 03:

```
1. struct PhanSo
2. {
3.     int Tu;
4.     int Mau;
5. };
6. typedef struct PhanSo PHANSO;
7. PHANSO c[10];
```

+ Trong ví dụ trên ta nói **c** là mảng một chiều có tối đa **10** phần tử. Mỗi phần tử trong mảng có kiểu dữ liệu là kiểu cấu trúc **PHANSO**.

KỸ THUẬT NHẬP MẢNG MỘT CHIỀU

Kỹ thuật nhập mảng một chiều

Vấn đề 1: Định nghĩa hàm nhập mảng một chiều số nguyên.

Kỹ thuật nhập mảng một chiều

— Vấn đề 1: Nhập mảng một chiều số nguyên.

— Khai báo hàm:

```
void Nhap(int [], int&);
```

+ Khai báo hàm **Nhap**.

+ Kiểu dữ liệu trả về không có (**void**).

+ Hàm **Nhap** có hai tham số đầu vào.

+ Tham số thứ nhất có kiểu dữ liệu là mảng một chiều các số nguyên.

+ Tham số thứ nhất là tham biến.

+ Tham số thứ hai có kiểu dữ liệu là **int**.

+ Tham số thứ hai là tham biến.

Kỹ thuật nhập mảng một chiều

```
void Nhap(int [], int&);
```

1. Khai báo hàm **Nhap**.
2. Kiểu dữ liệu trả về không có (**void**).
3. Hàm **Nhap** có hai tham số đầu vào.
4. Tham số thứ nhất có kiểu dữ liệu là mảng một chiều các số nguyên.
5. Tham số thứ nhất là tham biến.
6. Tham số thứ hai có kiểu dữ liệu là **int**.
7. Tham số thứ hai là tham biến.

Kỹ thuật nhập mảng một chiều

```
void Nhap(int [], int&);
```

Khai báo hàm Nhap

Kỹ thuật nhập mảng một chiều

```
11. void Nhap(int a[], int& n)
12. {
13.     cout << "Nhap n : ";
14.     cin >> n;
15.     for (int i = 0; i < n; i++)
16.     {
17.         cout << "Nhap a[" << i << "]: ";
18.         cin >> a[i];
19.     }
20. }
```

Định nghĩa hàm Nhap

Kỹ thuật nhập mảng một chiều

Vấn đề 2: Định nghĩa hàm nhập
mảng một chiều số thực.

Kỹ thuật nhập mảng một chiều

```
void Nhap(float [], int&);
```

Khai báo hàm Nhap

Kỹ thuật nhập mảng một chiều

```
void Nhap(float [], int&);
```

1. Khai báo hàm **Nhap**.
2. Kiểu dữ liệu trả về không có (**void**).
3. Hàm **Nhap** có hai tham số đầu vào.
4. Tham số thứ nhất có kiểu dữ liệu là mảng một chiều các số thực.
5. Tham số thứ nhất là tham biến.
6. Tham số thứ hai có kiểu dữ liệu là **int**.
7. Tham số thứ hai là tham biến.

Kỹ thuật nhập mảng một chiều

```
11. void Nhap(float a[], int& n)
12. {
13.     cout << "Nhap n : ";
14.     cin >> n;
15.     for (int i = 0; i < n; i++)
16.     {
17.         cout << "Nhap a[" << i << "]: ";
18.         cin >> a[i];
19.     }
20. }
```

Định nghĩa hàm Nhap

Kỹ thuật nhập mảng một chiều

Vấn đề 3: Định nghĩa hàm nhập mảng một chiều số nguyên dài.

Kỹ thuật nhập mảng một chiều

```
void Nhap(long [], int&);
```

Khai báo hàm Nhap

Kỹ thuật nhập mảng một chiều

```
void Nhap(long [], int&);
```

1. Khai báo hàm **Nhap**.
2. Kiểu dữ liệu trả về không có (**void**).
3. Hàm **Nhap** có hai tham số đầu vào.
4. Tham số thứ nhất có kiểu dữ liệu là mảng một chiều các số nguyên dài.
5. Tham số thứ nhất là tham biến.
6. Tham số thứ hai có kiểu dữ liệu là **int**.
7. Tham số thứ hai là tham biến.

Kỹ thuật nhập mảng một chiều

```
11. void Nhap(long a[], int& n)
12. {
13.     cout << "Nhap n : ";
14.     cin >> n;
15.     for (int i = 0; i < n; i++)
16.     {
17.         cout << "Nhap a[" << i << "]: ";
18.         cin >> a[i];
19.     }
20. }
```

Định nghĩa hàm Nhap

Kỹ thuật nhập mảng một chiều

Vấn đề 4: Định nghĩa hàm nhập mảng một chiều số thực dài.

Kỹ thuật nhập mảng một chiều

```
void Nhap(double [], int&);
```

Khai báo hàm Nhap

Kỹ thuật nhập mảng một chiều

```
void Nhap(double [], int&);
```

1. Khai báo hàm **Nhap**.
2. Kiểu dữ liệu trả về không có (**void**).
3. Hàm **Nhap** có hai tham số đầu vào.
4. Tham số thứ nhất có kiểu dữ liệu là mảng một chiều các số thực dài.
5. Tham số thứ nhất là tham biến.
6. Tham số thứ hai có kiểu dữ liệu là **int**.
7. Tham số thứ hai là tham biến.

Kỹ thuật nhập mảng một chiều

```
11. void Nhap(double a[], int& n)
12. {
13.     cout << "Nhap n : ";
14.     cin >> n;
15.     for (int i = 0; i < n; i++)
16.     {
17.         cout << "Nhap a[" << i << "]: ";
18.         cin >> a[i];
19.     }
20. }
```

Định nghĩa hàm Nhap

Kỹ thuật nhập mảng một chiều

Vấn đề 5: Định nghĩa hàm nhập mảng một chiều các phân số.

Kỹ thuật nhập mảng một chiều

```
void Nhap(PHANSO [], int&);
```

Khai báo hàm Nhap

Kỹ thuật nhập mảng một chiều

```
void Nhap(PHANSO [], int&);
```

1. Khai báo hàm **Nhap**.
2. Kiểu dữ liệu trả về không có (**void**).
3. Hàm **Nhap** có hai tham số đầu vào.
4. Tham số thứ nhất có kiểu dữ liệu là mảng một chiều các **PHANSO**.
5. Tham số thứ nhất là tham biến.
6. Tham số thứ hai có kiểu dữ liệu là **int**.
7. Tham số thứ hai là tham biến.

Kỹ thuật nhập mảng một chiều

```
11.struct PhanSo
12.{
13.|   int Tu;
14.|   int Mau;
15.};
16.typedef struct PhanSo PHANSO;
```

Khai báo kiểu dữ liệu cấu trúc PHANSO

Kỹ thuật nhập mảng một chiều

```
11. void Nhap(PHANSO& x)
12. {
13.     cout << "Nhap tu : ";
14.     cin >> x.Tu;
15.     cout << "Nhap mau : ";
16.     cin >> x.Mau;
17. }
```

Định nghĩa hàm Nhap PHANSO

Kỹ thuật nhập mảng một chiều

```
11. void Nhap(PHANSO a[], int& n)
12. {
13.     cout << "Nhap n : ";
14.     cin >> n;
15.     for (int i = 0; i < n; i++)
16.     {
17.         cout << "Nhap a[" << i << "]: ";
18.         Nhap(a[i]);
19.     }
20. }
```

Định nghĩa hàm Nhap

KỸ THUẬT XUẤT MẢNG MỘT CHIỀU

Kỹ thuật xuất một chiều

Vấn đề 1: Định nghĩa hàm xuất mảng một chiều số nguyên.

Kỹ thuật xuất một chiều

— Vấn đề 1: Xuất mảng một chiều số nguyên.

— Khai báo hàm:

```
void Xuat(int [], int);
```

+ Khai báo hàm **Xuat**.

+ Kiểu dữ liệu trả về không có (**void**).

+ Hàm **Xuat** có hai tham số đầu vào.

+ Tham số thứ nhất có kiểu dữ liệu là mảng một chiều các số nguyên.

+ Tham số thứ nhất là tham biến.

+ Tham số thứ hai có kiểu dữ liệu là **int**.

+ Tham số thứ hai là tham trị.

Kỹ thuật xuất một chiều

```
void Xuat(int [], int);
```

1. Khai báo hàm **Xuat**.
2. Kiểu dữ liệu trả về không có (**void**).
3. Hàm **Xuat** có hai tham số đầu vào.
4. Tham số thứ nhất có kiểu dữ liệu là mảng một chiều các số nguyên.
5. Tham số thứ nhất là tham biến.
6. Tham số thứ hai có kiểu dữ liệu là **int**.
7. Tham số thứ hai là tham trị.

Kỹ thuật xuất một chiều

```
void Xuat(int [], int);
```

Khai báo hàm Xuat

Kỹ thuật xuất một chiều

```
11. void Xuat(int a[], int n)
12. {
13.     for (int i = 0; i < n; i++)
14.         cout << setw(10) << a[i];
15. }
```

Định nghĩa hàm Xuat

Kỹ thuật xuất một chiều

Vấn đề 2: Định nghĩa hàm xuất mảng một chiều số thực.

Kỹ thuật xuất một chiều

```
void Xuat(float [], int);
```

Khai báo hàm Xuat

Kỹ thuật xuất một chiều

```
void Xuat(float [], int);
```

1. Khai báo hàm **Xuat**.
2. Kiểu dữ liệu trả về không có (**void**).
3. Hàm **Xuat** có hai tham số đầu vào.
4. Tham số thứ nhất có kiểu dữ liệu là mảng một chiều các số thực.
5. Tham số thứ nhất là tham biến.
6. Tham số thứ hai có kiểu dữ liệu là **int**.
7. Tham số thứ hai là tham trị.

Kỹ thuật xuất một chiều

```
11. void Xuat(float a[], int n)
12. {
13.     for (int i = 0; i < n; i++)
14.         cout << setw(10) << setprecision(3) << a[i];
15. }
```

Định nghĩa hàm Xuat

Kỹ thuật xuất một chiều

Vấn đề 3: Định nghĩa hàm xuất mảng một chiều số nguyên dài.

Kỹ thuật xuất một chiều

```
void Xuat(long [], int);
```

Khai báo hàm Xuat

Kỹ thuật xuất một chiều

```
void Xuat(long [], int);
```

1. Khai báo hàm **Xuat**.
2. Kiểu dữ liệu trả về không có (**void**).
3. Hàm **Xuat** có hai tham số đầu vào.
4. Tham số thứ nhất có kiểu dữ liệu là mảng một chiều các số nguyên dài.
5. Tham số thứ nhất là tham biến.
6. Tham số thứ hai có kiểu dữ liệu là **int**.
7. Tham số thứ hai là tham trị.

Kỹ thuật xuất một chiều

```
11. void Xuat(long a[], int n)
12. {
13.     for (int i = 0; i < n; i++)
14.         cout << setw(10) << a[i];
15. }
```

Định nghĩa hàm Xuat

Kỹ thuật xuất một chiều

Vấn đề 4: Định nghĩa hàm xuất mảng một chiều số thực dài.

Kỹ thuật xuất một chiều

```
void Xuat(doule [], int);
```

Khai báo hàm Xuat

Kỹ thuật xuất một chiều

```
void Xuat(double [], int);
```

1. Khai báo hàm **Xuat**.
2. Kiểu dữ liệu trả về không có (**void**).
3. Hàm **Xuat** có hai tham số đầu vào.
4. Tham số thứ nhất có kiểu dữ liệu là mảng một chiều các số thực dài.
5. Tham số thứ nhất là tham biến.
6. Tham số thứ hai có kiểu dữ liệu là **int**.
7. Tham số thứ hai là tham trị.

Kỹ thuật xuất một chiều

```
11. void Xuat(double a[], int n)
12. {
13.     for (int i = 0; i < n; i++)
14.         cout << setw(10) << setprecision(3) << a[i];
15. }
```

Định nghĩa hàm Xuat

Kỹ thuật xuất một chiều

Vấn đề 5: Định nghĩa hàm xuất mảng một chiều các phân số.

Kỹ thuật xuất một chiều

```
void Xuat(PHANSO [], int);
```

Khai báo hàm Xuat

Kỹ thuật xuất một chiều

```
void Xuat(PHANSO [], int);
```

1. Khai báo hàm **Xuat**.
2. Kiểu dữ liệu trả về không có (**void**).
3. Hàm **Xuat** có hai tham số đầu vào.
4. Tham số thứ nhất có kiểu dữ liệu là mảng một chiều các phân số (**PHANSO**).
5. Tham số thứ nhất là tham biến.
6. Tham số thứ hai có kiểu dữ liệu là **int**.
7. Tham số thứ hai là tham trị.

Kỹ thuật xuất một chiều

```
11.struct PhanSo
12.{
13.    int Tu;
14.    int Mau;
15.};
16.typedef struct PhanSo PHANSO;
```

Khai báo kiểu dữ liệu cấu trúc PHANSO

Kỹ thuật xuất một chiều

```
11. void Xuat(PHANSO x)
12. {
13. |   cout << x.Tu << "/" << x.Mau;
14. }
```

Định nghĩa hàm Nhập PHANSO

Kỹ thuật xuất một chiều

```
11. void Xuat(PHANSO a[], int n)
12. {
13.     for (int i = 0; i < n; i++)
14.         Xuat(a[i]);
15. }
```

Định nghĩa hàm Xuat



CHƯƠNG TRÌNH THỨ NHẤT

Chương trình thứ nhất

- Viết chương trình thực hiện các yêu cầu sau:
 - + Tạo ngẫu nhiên mảng một chiều các số nguyên với yêu cầu mỗi phần tử trong mảng là số nguyên nằm trong đoạn $[-100, 100]$.
 - + Xuất mảng ra màn hình.



Chương trình thứ nhất

```
11.#include <iostream>
12.#include <iomanip>
13.using namespace std;

14.void Nhap(int[], int&);
15.void Xuat(int[], int);
```

Khởi khai báo

Chương trình thứ nhất

```
16.int main()  
17.{  
18.    int b[500];  
19.    int k;  
20.    Nhap(b, k);  
21.    cout << "Mang ban dau:";  
22.    Xuat(b, k);  
23.    return 0;  
24.}
```

Định nghĩa hàm main()

Chương trình thứ nhất

```
25. void Nhap(int a[], int& n)
26. {
27.     cout << "Nhap n: ";
28.     cin >> n;
29.     srand(time(NULL));
30.     for (int i = 0; i < n; i++)
31.         a[i] = rand() % (200 + 1) - 100;
32. }
```

Định nghĩa hàm Nhap

Chương trình thứ nhất

```
33. void Xuat(int a[], int n)
34. {
35.     for (int i = 0; i < n; i++)
36.         cout << setw(10) << a[i];
37. }
```

Định nghĩa hàm Xuat



CHƯƠNG TRÌNH THỨ HAI

Chương trình thứ hai

- Viết chương trình thực hiện các yêu cầu sau:
 - + Tạo ngẫu nhiên mảng một chiều các số thực với yêu cầu mỗi phần tử trong mảng là số thực nằm trong đoạn $[-100, 100]$.
 - + Xuất mảng ra màn hình.



Chương trình thứ hai

```
11.#include <iostream>
12.#include <iomanip>
13.using namespace std;

14.void Nhap(float[], int&);
15.void Xuat(float[], int);
```

Khởi khai báo

Chương trình thứ hai

```
16.int main()  
17.{  
18.    float b[500];  
19.    int k;  
20.    Nhap(b, k);  
21.    cout << "Mang ban dau:";  
22.    Xuat(b, k);  
23.    return 1;  
24.}
```

Định nghĩa hàm main()

Chương trình thứ hai

```
25. void Nhap(float a[], int& n)
26. {
27.     cout << "Nhap n:";
28.     cin >> n;
29.     srand(time(NULL));
30.     for (int i = 0; i < n; i++)
31.         a[i] = -100.0 + (rand() /
32.             (RAND_MAX / (100.0 - (-100.0))));
33. }
```

Định nghĩa hàm Nhap

Chương trình thứ hai

```
34. void Xuat(float a[], int n)
35. {
36.     for (int i = 0; i < n; i++)
37.         cout << setw(10) << setprecision(3) << a[i];
38. }
```

Định nghĩa hàm Xuat



CHƯƠNG TRÌNH THỨ BA


Chương trình thứ ba

- Viết chương trình thực hiện các yêu cầu sau:
 - + Nhập mảng một chiều các số nguyên từ tập tin.
 - + Xuất mảng ra màn hình.



Chương trình thứ ba

- Định dạng tập tin `intdata01.inp`
 - + Dòng đầu tiên: số phần tử của mảng một chiều (n).
 - + Dòng tiếp theo: lưu n số nguyên tương ứng với các giá trị trong mảng một chiều các số nguyên.

 *intdata01.inp - Notepad
File Edit Format View Help
10
24 56 53 44 -54 6 63 -47 91 -99

Chương trình thứ ba

```
11.#include <iostream>
12.#include <fstream>
13.#include <iomanip>
14.using namespace std;

15.void Nhap(string, int[], int&);
16.void Xuat(int[], int);
```

Khởi khai báo

Chương trình thứ ba

```
17.int main()  
18.{  
19.    int b[500];  
20.    int k;  
21.    Nhap("intdata01.inp",b, k);  
22.    cout << "Mang ban dau:";  
23.    Xuat(b, k);  
24.    return 0;  
25.}
```

Định nghĩa hàm main()

Chương trình thứ ba

```
26. void Nhap(string filename, int a[], int& n)
27. {
    |
    |
    |
32. }
```

Định nghĩa hàm Nhap

Chương trình thứ ba

```
26. void Nhap(string filename, int a[], int& n)
27. {
28.     ifstream fi(filename);
32. }
```

Định nghĩa hàm Nhap

Chương trình thứ ba

```
26.void Nhap(string filename, int a[], int& n)
```

```
27.{
```

```
28.    ifstream fi(filename);
```

Ý 1: fi là đối tượng thuộc lớp ifstream.

```
32.}
```

Định nghĩa hàm Nhap

Chương trình thứ ba

```
26.void Nhap(string filename, int a[], int& n)
27.{
28.    ifstream fi(filename);
    Ý 2: Dòng lệnh số 28 khai báo đối tượng fi với
    đối số có tên filename và có kiểu string.
32.}
```

Định nghĩa hàm Nhap

Chương trình thứ ba

```
26. void Nhap(string filename, int a[], int& n)
```

```
27. {
```

```
28.     ifstream fi(filename);
```

```
32. }
```

Ý 3: Khi chương trình thực hiện tới dòng lệnh số 28. Đối tượng `fi` gọi thực hiện phương thức thiết lập với đối số `filename` có kiểu `string`. Phương thức thiết lập sẽ mở tập tin có tên lưu trong biến `filename`.

Định nghĩa hàm Nhap

Chương trình thứ ba

```
26. void Nhap(string filename, int a[], int& n)
27. {
28.     ifstream fi(filename);
29.     fi >> n;
30.
31.
32. }
```

Định nghĩa hàm Nhap

Chương trình thứ ba

```
26. void Nhap(string filename, int a[], int& n)
27. {
28.     ifstream fi(filename);
29.     fi >> n;
30.     for (int i = 0; i <= n - 1; i++)
31.
32. }
```

Định nghĩa hàm Nhap

Chương trình thứ ba

```
26. void Nhap(string filename, int a[], int& n)
27. {
28.     ifstream fi(filename);
29.     fi >> n;
30.     for (int i = 0; i <= n - 1; i++)
31.         fi >> a[i];
32. }
```

Định nghĩa hàm Nhap

Chương trình thứ ba

```
33. void Xuat(int a[], int n)
34. {
35.     for (int i = 0; i <= n - 1; i++)
36.         cout << setw(10) << a[i];
37. }
```

Định nghĩa hàm Xuat



CHƯƠNG TRÌNH THỨ TƯ

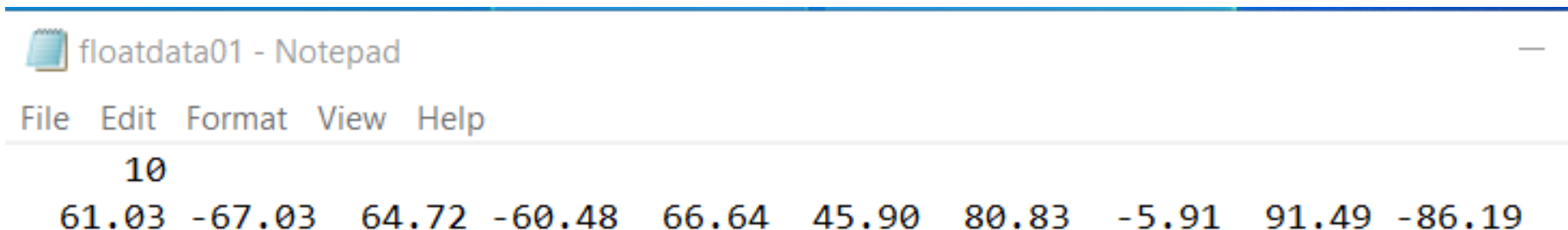
Chương trình thứ tư

- Viết chương trình thực hiện các yêu cầu sau:
 - + Nhập mảng một chiều các số thực từ tập tin.
 - + Xuất mảng ra màn hình.



Chương trình thứ ba

- Định dạng tập tin `floatdata01.inp`
 - + Dòng đầu tiên: số phần tử của mảng một chiều (n).
 - + Dòng tiếp theo: lưu n số thực tương ứng với các giá trị trong mảng một chiều các số thực.



```
floatdata01 - Notepad
File Edit Format View Help
10
61.03 -67.03 64.72 -60.48 66.64 45.90 80.83 -5.91 91.49 -86.19
```

Chương trình thứ tư

```
11.#include <iostream>
12.#include <fstream>
13.#include <iomanip>
14.using namespace std;

15.void Nhap(string, float[], int&);
16.void Xuat(float[], int);
```

Khởi khai báo

Chương trình thứ tư

```
17.int main()  
18.{  
19.    float b[500];  
20.    int k;  
21.    Nhap("floatdata01.inp",b, k);  
22.    cout << "Mang ban dau:";  
23.    Xuat(b, k);  
24.    return 0;  
25.}
```

Định nghĩa hàm main()

Chương trình thứ tư

```
26. void Nhap(string filename, float a[], int& n)
27. {
    |
    |
    |
32. }
```

Định nghĩa hàm Nhap

Chương trình thứ tư

```
26. void Nhap(string filename, float a[], int& n)
27. {
28.     ifstream fi(filename);
32. }
```

Định nghĩa hàm Nhap

Chương trình thứ tư

```
26. void Nhap(string filename, float a[], int& n)
27. {
28.     ifstream fi(filename);
        Ý 1: fi là đối tượng thuộc lớp ifstream.
32. }
```

Định nghĩa hàm Nhap

Chương trình thứ tư

```
26. void Nhap(string filename, float a[], int& n)
```

```
27. {
```

```
28.     ifstream fi(filename);
```

Ý 2: Dòng lệnh số 28 khai báo đối tượng fi với
đối số có tên filename và có kiểu string.

```
32. }
```

Định nghĩa hàm Nhap

Chương trình thứ tư

```
26. void Nhap(string filename, float a[], int& n)
```

```
27. {
```

```
28.     ifstream fi(filename);
```

Ý 3: Khi chương trình thực hiện tới dòng lệnh số 28. Đối tượng `fi` gọi thực hiện phương thức thiết lập với đối số `filename` có kiểu `string`. Phương thức thiết lập sẽ mở tập tin có tên lưu trong biến `filename`.

Định nghĩa hàm Nhap

Chương trình thứ tư

```
26. void Nhap(string filename, float a[], int& n)
27. {
28.     ifstream fi(filename);
29.     fi >> n;
30.
31.
32. }
```

Định nghĩa hàm Nhap

Chương trình thứ tư

```
26. void Nhap(string filename, float a[], int& n)
27. {
28.     ifstream fi(filename);
29.     fi >> n;
30.     for (int i = 0; i <= n - 1; i++)
31.
32. }
```

Định nghĩa hàm Nhap

Chương trình thứ tư

```
26. void Nhap(string filename, float a[], int& n)
27. {
28.     ifstream fi(filename);
29.     fi >> n;
30.     for (int i = 0; i <= n - 1; i++)
31.         fi >> a[i];
32. }
```

Định nghĩa hàm Nhap

Chương trình thứ tư

```
33. void Xuat(float a[], int n)
34. {
35.     for (int i = 0; i <= n - 1; i++)
36.         cout << setw(10) << setprecision(3) << a[i];
37. }
```

Định nghĩa hàm Xuat

Cảm ơn quý vị đã lắng nghe

ĐẠI HỌC QUỐC GIA TP.HCM
TRƯỜNG ĐH CÔNG NGHỆ THÔNG TIN
TOÀN DIỆN – SÁNG TẠO – PHỤNG SỰ