**Chương 1 – Những yếu tố cơ bản  của ngôn ngữ lập trình C**

**1- Các kiểu dữ liệu cơ sở**

char, int, long, float, double

**2- Khai báo biến**

<kiểu dữ liệu> <danh sách biến> ;

Chú ý: Vừa khai báo biến vừa khởi tạo giá trị ban đầu, VD: int   i = 0 ;

i = 3 ;

int a, b;

- Toán tử lấy địa chỉ biến &

**3- Khai báo mảng**

<kiểu dữ liệu> <tên mảng> [chỉ số 1]  [chỉ số 2] …

VD: int  x[100] ;  float  a[20] [10] ;

Chú ý: Định nghĩa kiểu dữ liệu mảng bằng toán tử typedef

VD:

typedef  int  vecto [100] ;

typedef  float  matran [20] [10] ;

Khai báo mảng có kiểu vừa định nghĩa ở trên:

vecto  x, y;

matran  a, b, c ;

**4- Các phép toán**

a/ Các phép toán **số học**: +, -, \*, /, %

Chú ý: Phép chia 2 toán hạng nguyên sẽ cho kết quả nguyên, VD: 1/2 cho kết quả là 0.

b/ Các phép toán **quan hệ**: ==, !=, <, <=, >, >= . Kết quả các phép quan hệ có thể là ≠ 0 (nghĩa là đúng), hoặc = 0 (nghĩa là sai).

c/  Các phép toán **logic**: &&, ||, !. Kết quả các phép quan hệ có thể là ≠ 0 (nghĩa là đúng), hoặc = 0 (nghĩa là sai).

d/ Lệnh **gán**: <biến>  =  <biểu thức> ;

e/ **Biểu thức điều kiện**: e1 ?  e2  :  e3 ;

f/ **Các phép tăng, giảm** 1:

n++ (dạng hậu tố)

++n (dạng tiền tố)

n - - (dạng hậu tố)

- - n (dạng tiền tố)

g/ **Chuyển đổi kiểu giá trị**

( <kiểu dữ liệu> ) <biểu thức>

VD: (long) sqrt (n)

float a = 5.8

int b;

b=a //incorrect

**------**

**Thực hành**

**Khung/định dạng cơ sở**

/\*

mentor: chienle

class: c programming

\*/

#include<stdio.h>

int main() {

//body

printf("Hello, World!");

return 0;

}

**Kiểu dữ liệu cơ sở**

|  |  |  |  |
| --- | --- | --- | --- |
| **Type** | **Storage size** | **Value range** | **Format** |
| char | 1 byte | -128 to 127 or 0 to 255 | %c |
| unsigned char | 1 byte | 0 to 255 | %c |
| signed char | 1 byte | -128 to 127 | %s |
| int | 2 or 4 bytes | -32,768 to 32,767 or -2,147,483,648 to 2,147,483,647 | %d |
| unsigned int | 2 or 4 bytes | 0 to 65,535 or 0 to 4,294,967,295 | %u |
| short | 2 bytes | -32,768 to 32,767 |  |
| unsigned short | 2 bytes | 0 to 65,535 |  |
| long | 4 bytes | -2,147,483,648 to 2,147,483,647 | %ld |
| unsigned long | 4 bytes | 0 to 4,294,967,295 |  |

**Nhập**

scanf("%d", &age);

**Xuất**

printf("%d", age);

**Các ký tự điều khiển**

* \n : Nhảy xuống dòng kế tiếp canh về cột đầu tiên.
* \t : Tab ngang.
* \r : Nhảy về đầu hàng, không xuống hàng.
* \a : Tiếng kêu bip.
* \\ : In ra dấu \
* \” : In ra dấu “
* \’ : In ra dấu ‘
* %%: In ra dấu %

**Số học**

| **STT** | **Toán tử** | **Tên** | **Ví dụ** | **Kết quả** |
| --- | --- | --- | --- | --- |
| 1 | + | Cộng | 3 + 4 | 7 |
| 2 | – | Trừ | 10.5 – 3 | 7.5 |
| 3 | \* | Nhân | 3 \* 2 | 6 |
| 4 | / | Chia | 6 / 3 | 2 |
| 5 | % | Chia lấy dư | 8 % 3 | 2 |

**Quan hệ**

| **STT** | **Toán tử** | **Tên** | **Ví dụ** | **Kết quả** |
| --- | --- | --- | --- | --- |
| 1 | == | So sánh bằng | 5 == 5 | 1 |
| 2 | != | So sánh khác | 5 != 5 | 0 |
| 3 | > | So sánh lớn hơn | 5 > 4 | 1 |
| 4 | < | So sánh nhỏ hơn | 5 < 4 | 0 |
| 5 | >= | So sánh lơn hơn hoặc bằng | 5 >= 4 | 1 |
| 6 | <= | So sánh nhỏ hơn hoặc bằng | 5 <= 5 | 1 |

**Luận lý**

| **STT** | **Toán tử** | **Tên** | **Ví dụ** | **Kết quả** |
| --- | --- | --- | --- | --- |
| 1 | ! | Phủ định | ! (5 > 4) | 0 |
| 2 | && | Và | 5 > 4 && 5 > 6 | 0 |
| 3 | || | Hoặc | 5 > 4 || 5 > 6 | 1 |

## Toán tử tăng giảm

#include <stdio.h>

int main()

{

    int i, k;

    i = 5; k = i++;

    printf("i = %d, k = %d\n", i, k);

    i = 5; k = ++i;

    printf("i = %d, k = %d\n", i, k);

    i = 5; k = i--;

    printf("i = %d, k = %d\n", i, k);

    i = 5; k = --i;

    printf("i = %d, k = %d\n", i, k);

    return 0;

}

i = 6, k = 5  
i = 6, k = 6  
i = 4, k = 5  
i = 4, k = 4

## Khai báo và gán

x = 5 => x = 5

c = x+ 5 = 5+5  
x += 5 => x = 10  
x -= 5 => x = 5  
x \* = 5 => x = 25  
x /= 5 => x = 5

**Cú pháp điều khiển**

// Nếu điều kiện 1 đúng thì  ...

if ( condition1) {

    // Làm gì đó khi điều kiện 1 đúng.

}

// Ngược lại nếu điều kiện 2 đúng thì ....

Else{

    // Làm gì đó khi điều kiện 2 đúng

    // (Điều kiện 1 sai).

}

// Ngược lại nếu điều kiện N đúng thì ...

else if( conditionN ) {

    // Làm gì đó khi điều kiện N đúng

    // (Các điều kiện ở trên sai).

}

// Các trường hợp còn lại.

else {

    // Làm gì đó ở đây.

}

/\*

mentor: chienle

class: c programming

\*/

#include<stdio.h>

int main() {

int age;

printf("Please enter your age: \n");

scanf("%d", &age);

if (age < 80) {

printf("You are pretty young");

}

else if (age >= 80 && age <= 100) {

printf("You are old");

}

else {

printf("You are verry old");

}

return 0;

}

**Vòng lặp**

// variable\_initialization: Khởi tạo một biến.

// condition: Điều kiện.

// variable\_update: Cập nhập giá trị mới cho biến.

for ( variable\_initialization; condition; variable\_update ) {

    // Các code cần thực thi khi điều kiện còn đúng.

}

for(int a= 0 ; a>5 ; a++)

{

}

/\*

mentor: chienle

class: c programming

\*/

#include<stdio.h>

int main() {

for(int x = 0; x < 5; x++)

{

printf("number: %d\n", x);

}

return 0;

}

**Chương 1 – Những yếu tố cơ bản  của ngôn ngữ lập trình C**

**1- Các kiểu dữ liệu cơ sở**

char, int, long, float, double

**2- Khai báo biến**

<kiểu dữ liệu> <danh sách biến> ;

Chú ý: Vừa khai báo biến vừa khởi tạo giá trị ban đầu, VD: int   i = 0 ;

i = 3 ;

int a, b;

- Toán tử lấy địa chỉ biến &

**3- Khai báo mảng**

<kiểu dữ liệu> <tên mảng> [chỉ số 1]  [chỉ số 2] …

VD: int  x[100] ;  float  a[20] [10] ;

Chú ý: Định nghĩa kiểu dữ liệu mảng bằng toán tử typedef

VD:

typedef  int  vecto [100] ;

typedef  float  matran [20] [10] ;

Khai báo mảng có kiểu vừa định nghĩa ở trên:

vecto  x, y;

matran  a, b, c ;

**4- Các phép toán**

a/ Các phép toán **số học**: +, -, \*, /, %

Chú ý: Phép chia 2 toán hạng nguyên sẽ cho kết quả nguyên, VD: 1/2 cho kết quả là 0.

b/ Các phép toán **quan hệ**: ==, !=, <, <=, >, >= . Kết quả các phép quan hệ có thể là ≠ 0 (nghĩa là đúng), hoặc = 0 (nghĩa là sai).

c/  Các phép toán **logic**: &&, ||, !. Kết quả các phép quan hệ có thể là ≠ 0 (nghĩa là đúng), hoặc = 0 (nghĩa là sai).

d/ Lệnh **gán**: <biến>  =  <biểu thức> ;

e/ **Biểu thức điều kiện**: e1 ?  e2  :  e3 ;

f/ **Các phép tăng, giảm** 1:

n++ (dạng hậu tố)

++n (dạng tiền tố)

n - - (dạng hậu tố)

- - n (dạng tiền tố)

g/ **Chuyển đổi kiểu giá trị**

( <kiểu dữ liệu> ) <biểu thức>

VD: (long) sqrt (n)

float a = 5.8

int b;

b=a //incorrect

h/ **Một số hàm toán học thông dụng**

**#include <math.h>**

fabs(x)

sin(x)

cos(x)

exp(x)

log(x)

log10(x)

sqrt(x)

pow(x, y)

5- **Nhập, xuất dữ liệu**

a/ Nhập:

- Lệnh scanf (“<xâu quy cách>” , <**danh sách** địa chỉ biến>);

b/ Xuất:

Lệnh printf (“<xâu quy cách>” , < danh sách biểu thức>) ;

Một số định dạng nhập - xuất dữ liệu:

%d              cho kiểu dữ liệu int

%ld             cho kiểu dữ liệu long

%f               cho kiểu dữ liệu float

%lf              cho kiểu dữ liệu double

%c              cho kiểu dữ liệu char

%s              cho kiểu dữ liệu xâu ký tự char\*

**Chương 2 – Các lệnh điều khiển**

**1- Lệnh đơn**: Lệnh scanf, printf, lệnh gán, lệnh ++, lệnh --, lời gọi hàm…

**2- Lệnh if**

if ( <biểu thức> )  <lệnh> ;

if ( <biểu thức> )  <lệnh 1 >  ;  else  <lệnh 2 > ;

if ( <biểu thức> )  <lệnh 1 >  ;   else if ( <biểu thức> )  <lệnh 1 >  ;   else  <lệnh 2 > ;

**3- Lệnh lặp for**

for ( <thành phần 1> ; <thành phần 2 > ; <thành phần 3> )  <lệnh> ;

VD: for (i=0; i < n; i++) s + = a[i] ;

**7 – Mảng**

**7.1. Mảng 1 chiều**

a. Khai báo mảng 1 chiều

Cách 1: Khai báo trực tiếp biến mảng

**<kiểu dữ liệu> <tên mảng> [ <số phần tử tối đa của mảng>] ;**

Cách 2: Khai báo một kiểu dữ liệu mới là mảng, sau đó sử dụng kiểu đó để khai báo biến.

Khai báo kiểu dữ liệu mảng 1 chiều sử dụng từ khóa **typedef** như sau:

**typedef <kiểu d/l> <tên kiểu d/l mảng> [ <số ph/tử tối đa của mảng>] ;**

Sử dụng kiểu dữ liệu mảng trên để khai báo biến mảng như sau:

**<tên kiểu dữ liệu mảng>  <tên mảng> ;**

b. Truy xuất phần tử mảng 1 chiều

**<tên mảng> [ <chỉ số của phần tử> ]**

**7.2. Mảng 2 chiều**

a. Khai báo mảng 2 chiều

- Khai báo trực tiếp mảng 2 chiều:

**<kiểu dữ liệu> <tên mảng> [ <số dòng tối đa>] [<số cột tối đa>] ;**

- Khai báo kiểu mảng 2 chiều, sau đó sử dụng kiểu này để khai báo mảng :

**typedef <kiểu dữ liệu> <tên kiểu mảng> [ <số dòng tối đa>] [<số cột tối đa>] ;**

Khai báo mảng :

<tên kiểu mảng>  <tên mảng>

b. Truy xuất phần tử mảng 2 chiều

**<tên  mảng> [<chỉ số dòng>] [<chỉ số cột>]**

**7.3. Các thao tác trên mảng**

- Nhập – Xuất mảng

- Tìm kiếm trên mảng: Tìm max, min, tìm những phần tử thoả mãn điều kiện nào đó; tìm phần tử bằng giá trị x nào đó, tìm vị trí đầu tiên v.v...

- Đếm các phần tử thoả mãn điều kiện nào đó.

- Tính tổng, trung bình cộng ... các phần tử mảng.

- Đổi chỗ 2 phần tử mảng.

- Đổi chỗ 2 hàng (hoặc 2 cột) ma trận

- Sắp xếp tăng (hoặc giảm).

- Chèn thêm phần tử; chèn thêm hàng, cột ma trận.

- Xoá bớt phần tử; xóa hàng (hoặc cột) ma trận.

**Chương 4 – Hàm**

1- Khai báo hàm (nguyên mẫu hàm)

[<kiểu trả về>] <tên hàm> ( [<danh sách kiểu tham số hình thức>] );

**2- Định nghĩa hàm (xây dựng hàm)**

[<kiểu trả về>] <tên hàm> ( [<danh sách tham số hình thức>] )

{

<Các khai báo biến, mảng ...>

<Các câu lệnh>

}

3- Gọi hàm (sử dụng hàm)

<tên hàm> ([<danh sách tham số thực sự>])

**Thực hành**

1. Viết chương trình  
a. Kiểm tra số chẵn lẻ nhập từ bàn phím  
b. Nhập 2 số từ bàn phím, nếu số thứ nhất lớn hơn hoặc bằng số thứ hai thì kiểm tra   
Lớn hơn hoặc bằng; ngược lại hiển hiển thị nhỏ hơn.  
INPUT  
A>=B  
------(A>B)  
------------- Show A>B  
------Show A=B  
Show A<B  
c. Tính tổng của a = 3, b= 4.5, c=5.2  
d. Dùng mảng, tính điểm trung bình từ của 80, 12 và 20. Hiển thị giá trị phần tử thứ 2  
e. Dùng mảng tính cấp số nhân của 1 dãy số.  
Ví dụ: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10. Kết quả = 3628800

1. // Check whether an integer is odd or even
2. #include <stdio.h>
3. int main()
4. {
5. int number;
6. printf("Enter an integer: ");
7. scanf("%d", &number);
8. // True if the remainder is 0
9. if (number%2 == 0)
10. {
11. printf("%d is an even integer.",number);
12. }
13. else
14. {
15. printf("%d is an odd integer.",number);
16. }
17. return 0;
18. }

OUTPUT

Enter an integer: 7

7 is an odd integer.

1. if (test expression1)
2. {
3. // statement(s)
4. }
5. else if(test expression2)
6. {
7. // statement(s)
8. }
9. else if (test expression3)
10. {
11. // statement(s)
12. }
13. .
14. .
15. else
16. {
17. // statement(s)
18. }

------------

1. // Program to relate two integers using =, > or < symbol
2. #include <stdio.h>
3. int main()
4. {
5. int number1, number2;
6. printf("Enter two integers: ");
7. scanf("%d %d", &number1, &number2);
8. //checks if the two integers are equal.
9. if(number1 == number2)
10. {
11. printf("Result: %d = %d",number1,number2);
12. }
13. //checks if number1 is greater than number2.
14. else if (number1 > number2)
15. {
16. printf("Result: %d > %d", number1, number2);
17. }
18. //checks if both test expressions are false
19. else
20. {
21. printf("Result: %d < %d",number1, number2);
22. }
23. return 0;
24. }

OUT PUT

Enter two integers: 12

23

Result: 12 < 23

#include <stdio.h>

int main() {

int a = 3;

float b = 4.5;

double c = 5.25;

float sum;

sum = a + b + c;

printf("The sum of a, b, and c is %f.", sum);

return 0;

}

#include <stdio.h>

int main() {

/\* TODO: define the grades variable here \*/

int grades[3];

int average;

grades[0] = 80;

/\* TODO: define the missing grade

so that the average will sum to 85. \*/

grades[1] = 85;

grades[2] = 90;

average = (grades[0] + grades[1] + grades[2]) / 3;

printf("The average of the 3 grades is: %d", average);

return 0;

}

#include <stdio.h>

int main() {

int array[] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };

int factorial = 1;

int i;

for(i=0;i<10;i++){

factorial \*= array[i];

}

printf("10! is %d.\n", factorial);

}

----------

CON TRỎ

#include <stdio.h>

int main()

{

int c = 5;

int \*pc = &c;

// int \*pc;

// pc = &c;

// int\* pc = &c;

printf("%d", \*pc); // 5

}

#include <stdio.h>

int main()

{

int\* pc, c;

c = 22;

printf("Address of c: %p\n", &c);

printf("Value of c: %d\n\n", c); // 22

pc = &c;

printf("Address of pointer pc: %p\n", pc);

printf("Content of pointer pc: %d\n\n", \*pc); // 22

c = 11;

printf("Address of pointer pc: %p\n", pc);

printf("Content of pointer pc: %d\n\n", \*pc); // 11

\*pc = 2;

printf("Address of c: %p\n", &c);

printf("Value of c: %d\n\n", c); // 2

return 0;

}

OUTPUT

Address of c: 2686784

Value of c: 22

Address of pointer pc: 2686784

Content of pointer pc: 22

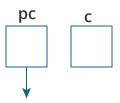
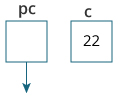
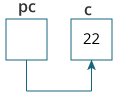
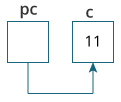
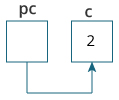
Address of pointer pc: 2686784

Content of pointer pc: 11

Address of c: 2686784

Value of c: 2

**GIẢI THÍCH**

1. int\* pc, c;  
    Một biến con trỏ và một biến bình thường được tạo ra.
2. c = 22;  
     
   22 được gán cho biến c.
3. pc = &c;  
     
   Địa chỉ của biến c được gán cho con trỏ pc.
4. c = 11;  
     
   11 được gán cho biến c.
5. \*pc = 2;  
     
   Thay đổi giá trị tại vị trí bộ nhớ được trỏ bởi con trỏ pc thành 2.

#include <stdio.h>

typedef struct {

char \* name;

int age;

} person;

int main() {

person chienle;

chienle.name = "chienle";

chienle.age = 29;

printf("%s is %d years old.", chienle.name, chienle.age);

}

#include <stdio.h>

#include <stdlib.h>

struct sinhvien

{

char MSV[20];

char hoten[30];

double toan, tin, anh;

};

int main()

{

struct sinhvien sv1, sv2;

printf("Nhap du lieu cho sv1:\n");

printf("MSV: "); fflush(stdin);

gets(sv1.MSV);

printf("Ho ten: "); fflush(stdin);

gets(sv1.hoten);

printf("Diem toan, tin, anh: "); fflush(stdin);

scanf("%lf %lf %lf", &sv1.toan, &sv1.tin, &sv1.anh);

printf("Nhap du lieu cho sv2:\n");

printf("MSV: "); fflush(stdin);

gets(sv2.MSV);

printf("Ho ten: "); fflush(stdin);

gets(sv2.hoten);

printf("Diem toan, tin, anh: "); fflush(stdin);

scanf("%lf %lf %lf", &sv2.toan, &sv2.tin, &sv2.anh);

printf("\n --------- Thong tin sinh vien -----\n");

printf("%-20s %-30s %-7s %-7s %-7s\n", "MSV", "Ho ten", "Toan", "Tin", "Anh");

printf("%-20s %-30s %-7.2lf %-7.2lf %-7.2lf\n", sv1.MSV, sv1.hoten, sv1.toan, sv1.tin, sv1.anh);

printf("%-20s %-30s %-7.2lf %-7.2lf %-7.2lf\n", sv2.MSV, sv2.hoten, sv2.toan, sv2.tin, sv2.anh);

return 0;

}

#include <stdio.h>

typedef struct {

int x;

int y;

} point;

int main() {

point A;

A.x = 10;

A.y = 5;

printf("x = %d\ny = %d",A.x,A.y);

}

#include <stdio.h>

typedef struct{

int x;

int y;

} Point1;

typedef struct {

Point1 a;

Point1 b;

Point1 c;

} Triangle1;

int main() {

Triangle1 tg;

tg.a.x = 5;

tg.a.y = 20;

tg.b.x = 10;

tg.c.y = 20;

printf("tg.A.x = %d",tg.a.x);

}

#include <stdio.h>

struct person

{

int age;

float weight;

};

int main()

{

struct person \*personPtr, person1;

personPtr = &person1;

printf("Enter age: ");

scanf("%d", &personPtr->age);

printf("Enter weight: ");

scanf("%f", &personPtr->weight);

printf("Displaying:\n");

printf("Age: %d\n", personPtr->age);

printf("weight: %f", personPtr->weight);

return 0;

}

|  |  |
| --- | --- |
| 2  3  4  5  6  7 | typedef struct  {      dataType member1;      dataType member2;      ...  } structureName; |

-----

Một số dạng khai báo cấu trúc

//#include <conio.h>

#include <stdio.h>

//#include <string.h>

//Tao cau truc Thisinh

typedef struct {

char sbd[5], hoten[20];

float diem1, diem2, diem3;

} Thisinh;

//Khai bao kieu Mang gom 100 phan tu kieu Thisinh

typedef Thisinh Mang[100];

//chuong trinh con Nhap

void nhap(Mang m, int &n){

int i;

//trung gian nhap du lieu

//cho cac thanh phan khong nguyen trong kieu cau truc

float tg;

printf("Bao nhieu thi sinh: ");

scanf("%d", &n);

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

fflush(stdin);

printf("Thi sinh [%d]\n", i+1);

printf("SBD: ");

gets(m[i].sbd);

fflush(stdin);

printf("Ho ten: ");

gets(m[i].hoten);

printf("Diem 1:");

scanf("%f", &tg);

m[i].diem1=tg;

printf("Diem 2:");

scanf("%f", &tg);

m[i].diem2=tg;

printf("Diem 3:");

scanf("%f", &tg);

m[i].diem3=tg;

}

}

//Chuong trinh con hien

void hien(Mang m, int n){

int i;

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

printf("\n %5s%21s%5.1f%5.1f%5.1f",

m[i].sbd, m[i].hoten, m[i].diem1, m[i].diem2, m[i].diem3);

}

}

int main(){

Mang a;

int n, i;

nhap(a, n);

hien(a, n);

//getch();

return 0;

}

Đọc ghi file

Bảo mật, lớn VS văn bản thường

/\*

FILE

Doc, ghi file nhi phan

size\_t fread(void \*ptr size\_t size, size\_t count, FILE \*stream)

size\_t fwrite(const void \*ptr, size\_t size, size\_t count, FILE \*stream);

\*/

#include <stdio.h>

typedef struct{

char hoten[30];

int tuoi;

} Nguoi;

int main(){

Nguoi a = {"chienle", 18};

Nguoi b;

FILE\* f;

//strcpy(a.hoten,"Nguyen Van Hoa");

//a.tuoi=20;

//Mo file de ghi

f=fopen("binary.bin", "wb");

fwrite(&a, sizeof(a), 1, f);

fclose(f);

//Mo file de doc

f=fopen("binary.bin","rb");

fread(&b, sizeof(b), 1, f);

fclose(f);

printf("\n Ho ten: %s", b.hoten);

printf("\n Tuoi: %d", b.tuoi);

return 0;

}

#include <stdio.h>

int main()

{

int num;

FILE \*fptr;

//fptr = fopen("C:\\program.txt","w");

fptr = fopen("text.txt", "w");

if(fptr == NULL)

{

printf("Error!");

}

printf("Enter Interger: ");

scanf("%d", &num);

fprintf(fptr,"%d",num);

fclose(fptr);

return 0;

}

-----

#include <stdio.h>

int main()

{

int num;

FILE \*fptr;

if ((fptr = fopen("text1.txt","r")) == NULL){

printf("Error! opening file");

}

fscanf(fptr,"%d", &num);

printf("Value of n=%d", num);

fclose(fptr);

return 0;

}

Cấu trúc dữ liệu và các phép toán trên mảng – array

a. khai báo và nhập danh sách số nguyên từ bàn phím (khai báo mảng động)

b. in danh sách số nguyên vừa nhập ở câu a

c. tính tổng dãy số vừa nhập ở câu a

d. sắp xếp giảm dần từ danh sách trên

e. tìm số nguyên lớn nhất từ dãy số trên

arr = {2,5,1,3,7,11,9}

A[i] = {11,9,7,3,7,11}

B[i] = {1,1,2,3,5,5}

For i = 5

fori - 5

forj - 3

forj - 6

Max6

A[i] = {6 3}

B[i] = {5 3}

fori - 3

forj - 3

forj - 5

Max5