HOA SEN UNIVERSITY

www.hoasen.edu.vn

Cấu Trúc Dữ Liệu & Giải Thuật

Data Structures & Algorithms

GV: Phan Hồng Trung

HOA SEN UNIVERSITY

www.hoasen.edu.vn

Bài 1 – Ôn Tập

- Tham số dòng lệnh (Command line parameters)
- Nhập/Xuất dữ liệu (Data I/O)
- Hàm (Functions)
- Con trỏ (Pointers)
- Dãy/Mảng (Arrays)
- Cấu trúc (Structures)
- Đọc/Ghi tập tin (File I/O)

HOA SEN UNIVERSITY

www.hoasen.edu.vn

Hàm main() & tham số dòng lệnh

- Không sử dụng tham số dòng lệnh:
  - int main(){/*…*/}
- Sử dụng tham số dòng lệnh:
  - int main(int argc, char *argv[]){/*...*/}
  - int main(int argc, char **argv){/*...*/}
  - argc: (ARGument Count) số tham số dòng lệnh.
  - argv: (ARGument Vector) array chứa danh sách tham số.
  - argv[0] là tên của chương trình.
- Vd: liệt kê tham số dòng lệnh.
- Vd: tính tổng các tham số được cung cấp thông qua tham số dòng lệnh.

HOA SEN UNIVERSITY

www.hoasen.edu.vn

Hàm

- Định nghĩa hàm:
  - return_type *functionName*(*parameter*$_1$, …, *parameter*$_n$) {
    // code to be executed
    }
  - Dùng lệnh return để trả về giá trị của hàm.
  - Hàm không trả về giá trị: khai báo return_type là void.
- Ý nghĩa:
  - Định nghĩa hàm một lần, dùng lại nhiều lần.
  - Giúp viết chương trình rõ ràng, dễ hiểu.
  - Dễ bảo trì chương trình.

HOA SEN UNIVERSITY

www.hoasen.edu.vn

Hàm

- Vd: tính tổng sau:
  - $s = 1 + \frac{1+2}{2!} + \frac{1+2+3}{3!} + \cdots + \frac{1+2+\cdots+n}{n!}$
  - $s = -1 + \frac{1+2}{2!} - \frac{1+2+3}{3!} + \cdots (+/-) \frac{1+2+\cdots+n}{n!}$
- Vd: tính tổng các số nguyên tố ≤ n.
- Có 2 cách truyền tham số cho hàm:
  - Truyền bằng giá trị (Pass by Value): khi tham số hình thức thay đổi thì tham số thực không thay đổi.
  - Truyền bằng tham chiếu (Pass by Reference): khi tham số hình thức thay đổi thì tham số thực thay đổi theo.

HOA SEN UNIVERSITY

www.hoasen.edu.vn

Hàm

- Function overloading:
  - Nhiều hàm cùng tên nhưng khác tham số (kiểu dữ liệu, thứ tự, số lượng).
  - Khi gọi hàm, dựa vào kiểu dữ liệu của tham số thực máy tính sẽ chọn hàm phù hợp.
- Vd:
  - int add(int a, int b) ← add(2, 3)
  - float add(float a, float b) ← add(2.5f, 3.5f)
  - double add(double a, double b) ← add(2.5, 3.5)

HOA SEN UNIVERSITY

www.hoasen.edu.vn

Hàm

- Tham số mặc định:
  - Là tham số được gán giá trị mặc định lúc định nghĩa hàm.
  - Giúp giảm Function Overloading.
- Vd: hàm sau có b là tham số mặc định có giá trị mặc định là 0.
  - int sum(int a, int b = 0){
    return a+b;
    }
  - sum(2,3) → 5
  - sum(2) → 2

HOA SEN UNIVERSITY

www.hoasen.edu.vn

Con trỏ

- Con trỏ là biến lưu trữ địa chỉ bộ nhớ.
- Khai báo:
  - data_type* var_name;
- VD:

```
string food = "Pizza";// A food variable of type string
string* ptr = &food;// A pointer variable, with the name ptr, that stores the address of food

// Output the value of food (Pizza)
cout << food << "\n";
cout << *ptr << "\n";

// Output the memory address of food (0x6dfed4)
cout << &food << "\n";
cout << ptr << "\n";
```

HOA SEN UNIVERSITY

www.hoasen.edu.vn

Dãy một chiều

- Dãy là một dãy các phần tử cùng kiểu, được lưu trữ liên tục trong bộ nhớ máy tính, có kích thước cố định.
- Khai báo dãy một chiều:
  - type arrayName [arraySize];
  - double balance[10];
- Kết hợp khai báo và gán giá trị cho các phần tử của dãy:
  - double balance[] = {1000.0, 2.0, 3.4, 17.0, 50.0};

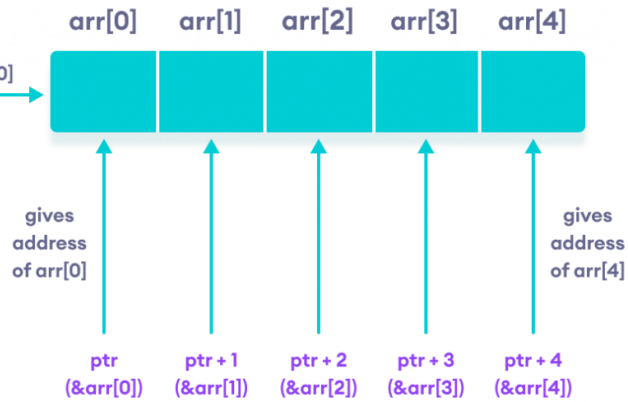
HOA SEN UNIVERSITY

www.hoasen.edu.vn

Dãy một chiều

- Con trỏ & dãy:
  - int *ptr;
    int arr[5];
    ptr = arr;

    ptr = arr → same as &arr[0]
  - ptr == arr == &arr[0];
    ptr + 1 == &arr[1];
    ptr + 2 == &arr[2];
    ptr + 3 == &arr[3];
    ptr + 4 == &arr[4];
  - *ptr == *arr == arr[0];
    *(ptr + 1) == arr[1];
    *(ptr + 2) == arr[2];
    *(ptr + 3) == arr[3];
    *(ptr + 4) == arr[4];



HOA SEN UNIVERSITY

www.hoasen.edu.vn

Dãy một chiều

- Tính số phần tử của dãy:
  - int arrSize = sizeof(arr)/sizeof(*arr);
  - int arrSize = end(arr) - begin(arr);
  - int arrSize = *(&arr + 1) - arr;
  - *(&arr + 1): địa chỉ bộ nhớ ngay sau phần tử cuối cùng của dãy.

HOA SEN UNIVERSITY

www.hoasen.edu.vn

Dãy hai chiều

- Tạo dãy hai chiều:
  - int x[3][4];
- Khởi tạo dãy hai chiều:
  - int test[2][3] = {{2, 4, 5}, {9, 0, 19}};

	Col 1	Col 2	Col 3	Col 4
Row 1	x[0][0]	x[0][1]	x[0][2]	x[0][3]
Row 2	x[1][0]	x[1][1]	x[1][2]	x[1][3]
Row 3	x[2][0]	x[2][1]	x[2][2]	x[2][3]

	Col 1	Col 2	Col 3
Row 1	2	4	5
Row 2	9	0	19

HOA SEN UNIVERSITY

www.hoasen.edu.vn

Cấu trúc

- Chứa tập hợp các phần tử có thể khác kiểu dữ liệu.
- Khai báo:

```
struct Person {
    char name[50];
    int age;
    float salary;
};
```

- Sử dụng:

```
Person p;
cin.get(p.name, 50);
cin >> p.age;
cin >> p.salary;
```


HOA SEN UNIVERSITY

www.hoasen.edu.vn

Đọc/Ghi tập tin

- Thư viện fstream cung cấp 3 class làm việc với file:
  - ifstream: (input file stream) đọc file.
  - ofstream: (output file stream) ghi file.
  - fstream: (file stream) đọc/ghi file.
- Để truy cập file:
  - Mở file: kết hợp file với các lớp stream.
  - Đọc/Ghi file.
  - Đóng file: giải phóng tài nguyên được sử dụng trong quá trình truy cập file.

HOA SEN UNIVERSITY

www.hoasen.edu.vn

Đọc/Ghi tập tin

- Mở file: open(filename, mode);

Mode	Mô tả
ios::in	Mở file để đọc (input)
ios::out	Mở file để ghi (output)
ios::binary	Mở file nhị nhân
ios::ate	Chuyển con trỏ file về cuối file sau khi mở file (at the end)
ios::app	Con trỏ file luôn ở cuối file, không thể di chuyển sang vị trí khác, dùng để ghi thêm dữ liệu vào cuối file (append)
ios::trunc	Xóa sạch nội dung cũ của file sau khi mở file (truncate)

HOA SEN UNIVERSITY

www.hoasen.edu.vn

Đọc/Ghi tập tin

- VD: mở file nhị phân để ghi thêm vào cuối file:
  - ofstream file;
    file.open("example.bin", ios::out | ios::app | ios::binary);
- Mode mở file mặc định của các lớp stream:

Class	Default mode
ofstream	ios::out
ifstream	ios::in
fstream	ios::in | ios::out

HOA SEN UNIVERSITY

www.hoasen.edu.vn

Đọc/Ghi tập tin

- Con trỏ stream:
  - get pointer: xác định vị trí đọc file.
  - put pointer: xác định vị trí ghi file.
- Lấy vị trí con trỏ: fs.tellg()/fs.tellp()
- Di chuyển con trỏ: fs.seekg()/fs.seekp()
  - fs.seekg(offset, reference position)
  - fs.seekp(offset, reference position)
  - fs.seekg(0, ios::beg);
  - fs.seekg(0, ios::end);

Reference Position	Mô tả
ios::beg	the beginning of the stream
ios::cur	the current position
ios::end	the end of the stream