|  |  |
| --- | --- |
| **TRƯỜNG ĐẠI HỌC THƯƠNG MẠI**  **HỌC KỲ I NĂM HỌC 2021 – 2022** | |
| ***(Phần dành cho sinh viên/ học viên)*** | |
| **Bài thi học phần:**  **Cấu trúc dữ liệu và giải thuật**  **Mã số đề thi: 1**  **Ngày thi:** **21/12/2021**  **Tổng số Trang: 10** | **Số báo danh: 61**  **Lớp: 2154INFO1311**  **Họ và tên: Nguyễn Thị Thiện** |
| **Điểm kết luận:** | **GV chấm thi 1:** ……………………......  **GV chấm thi 2:**….……………………. |

**Câu 1:Trình bày về khái niệm,đặc điểm và các phép toán đối với cấu trúc mảng**

*Trả lời:*

* ***Khái niệm:***

Mảng là một tập có thứ tự gồm một số cố định các phần tử cùng kiểu dữ liệu .Vậy mảng là một cấu trúc dữ liệu. Trong đó, mỗi phần tử được xác định và truy cập bởi tên mảng và chỉ số của phần tử đó (là thứ tự của phần tử) – đây là truy cập ngẫu nhiên.

Kiểu dữ liệu mảng: một chiều, hai chiều, ba chiều,… nhiều chiều.

*Ví dụ:* Vécto là mảng một chiều mỗi phần tử của nó ứng với chỉ số i. Ma trận là mảng hai chiều, mỗi phần tử của nó ứng với hai chỉ số i và j.

Mảng được sử dụng trong các trường hợp như: có nhiều phần tử; các phần tử có cùng đặc điểm.

*Ví dụ:* chương trình quản lý danh sách sinh viên => sử dụng mảng sinh viên, chương trình quản lý danh sách nhân viên => sử dụng mảng nhân viên,…

Cấu trúc lưu trữ mảng: là hình thức lưu trữ kế tiếp

- Địa chỉ các phần tử liên tiếp nhau

- Các phần tử được sắp xếp theo hàng

- Bộ nhớ cố định

* ***Đặc điểm***
* Cấu trúc đơn giản, truy cập nhanh (so với danh sách liên kết, việc truy cập đến một phần tử trong mảng nhanh hơn).
* Mảng chỉ thực sự hoạt động tốt với mảng tối đa 3 chiều
* Thiếu mềm dẻo trong phép toán như xóa, chèn. Ví dụ: Để xoá một phần tử không phải là phần tử cuối thì sử dụng cấu trúc mảng không hiệu quả. Bởi vì công việc này cần tốn thời gian cho việc dịch chuyển các phần tử còn lại lấp vào chỗ trống của mảng.
* Khi đã khai báo một mảng thì kích thước của nó là cố định, không thể thay đổi trong quá trình thực hiện chương trình -> Bất tiện khi chưa biết trước số phần tử cần xử lý. Nếu khai báo mảng lớn thì gây tốn bộ nhớ, ảnh hưởng đến hiệu suất chương trình; khai báo mảng nhỏ có thể thiếu bộ nhớ.
* Tất cả các phần tử của mảng đều có cùng kiểu dữ liệu.
* Một mảng cần có giới hạn số phần tử mà mảng có thể chứa.
* Phải cấp phát vùng nhớ mới có thể sử dụng mảng.
* Các phần tử trong mảng được cấp phát liên tiếp trong bộ nhớ máy tính.
* Chỉ số của mảng bắt đầu từ 0 và kết thúc là size - 1. Trong ví dụ ở phần trên, mảng player gồm 11 phần tử với các chỉ số là: 0, 1, 2,..., 10. - Để truy cập vào một phần tử của mảng có chỉ số là idx, bạn sử dụng cặp dấu [] với chỉ số ở giữa. Trong ví dụ trên, mảng player gồm các phần tử có giá trị là: player[0], player[1], player[2],..., player[10].
* Không thể so sánh 2 mảng với nhau. Dù cho chúng có cùng kích thước, giá trị của các phần tử giống nhau, nhưng chúng cũng không bằng nhau.
* Không thể gán giá trị của cả mảng một lúc, mà phải gán giá trị cho từng phần
* Các phần tử trong mảng dùng chung một tên và được truy xuất thông qua chỉ số phần tử.
* ***Các phép toán đối với cấu trúc mảng***
* Tạo lập mảng

Input: kích thước của mảng,kiểu dữ liệu của từng phần tử

Output: tạo ra mảng có kích thước và kiểu dữ liệu tương ứng cho từng phần tử

Ví dụ: tạo mạng số thực có số lượng phần tử là 50: float a[50];

* Nhập / xuất mảng

+ Nhập mảng

Input: mảng chưa có các giá trị/ các giá trị được mặc định, các giá trị cần nhập vào mảng

Output: mảng đã có giá trị được nhập vào

Thuật toán: dùng vòng lặp for với biến chạy i từ phần tử đầu tiên đến phần tử cuối cùng của mảng, trong đó mỗi i thực hiện nhập dữ liệu cho phần tử thứ i của mảng.

Ví dụ:

*int a [50];*

*for (int i = 0; i < 50 ; i++)*

*scanf (“%d”, &a[i]);*

+ Xuất mảng

Input: dữ liệu của mảng

Output: thông tin dữ liệu của mảng được hiển thị lên màn hình

Thuật toán: dùng vòng lặp for với biến chạy i từ phần tử đầu tiên đến phần tử cuối cùng. Với mỗi i thực hiện in giá trị của phần tử thứ i trong mảng ra màn hình

Ví dụ: với mảng số nguyen

*void xuat ( int a[50], int n)*

*{*

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

*{*

*printf("%4d", a[i]);*

*}*

*}*

* Cập nhật phần tử mảng

Hoạt động cập nhật là update giá trị của phần tử đang tồn tại trong mảng tại chỉ mục đã cho.

* Duyệt mảng (tìm kiếm một phần tử của mảng)

Input: mảng các phần tử

Output: thông tin cần tìm

Thuật toán: dùng vòng lặp for với biến chạy; từ phần tử đầu tiên đến phần tử cuối cùng, vỡi mỗi i thực hiện kiểm tra điều kiện

*Ví dụ:* Tính trung bình cộng của các số nguyên chẵn trong mảng

*int tbc(int a[50], int n)*

*{*

*int sum=0,dem=0;*

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

*{*

*if (a[i] <0)*

*{*

*sum = sum + a[i];*

*dem++;*

*}*

*}*

*float tb= sum / dem;*

*return tb;*

*}*

* Phép toán thêm bớt phần tử

+ Thêm phần tử

Hoạt động chèn là để chèn một hoặc nhiều phần tử dữ liệu vào trong một mảng. Tùy theo yêu cầu, phần tử mới có thể được chèn vào vị trí đầu, vị trí cuối hoặc bất kỳ vị trí chỉ mục đã cho nào của mảng.

* Bước 1: Lấy giá trị vị trí positon, giá trị value của phần tử cần truyền từ bàn phím.
* Bước 2: Kiểm tra vị trí chèn có phù hợp. Nếu < 0 thì thông báo không hợp lệ, Nếu > n thì thêm vào sau vị trí cuối cùng của mảng.
* Bước 3: Dịch chuyển toàn bộ các phần tử trong mảng về sau 1 đơn vị
* Bước 4: Gán giá trị value vào vị trí position.
* Bước 5: Tăng thêm n 1 đơn vị (biến lưu kích thước của mảng)

+ Xóa phần tử

Hoạt động xóa là xóa một phần tử đang tồn tại từ một mảng và tổ chức lại các phần tử còn lại trong mảng đó.

* Bước 1. Kiểm tra tính hợp lệ của vị trí position truyền vào.
* Bước 2. Dịch mảng về phái trước 1 đơn vị
* Bước 3. Giảm giá trị n (biến lưu kích thước mảng) đi 1

**Câu 2: Hãy dùng cấu trúc nhị phân tìm kiếm để xây dựng một chương trình quản lý một tập hợp các phần tử với giá trị khóa là các số nguyên. Chương trình cho phép:**

* **Tạo cây nhị phân tìm kiếm với n phần tử**
* **Thêm nút mới có giá trị v vào cây**
* **Đếm số nút là của cây**
* **Tính chiều cao của cây**
* **Hiển thị các giá trị nút trong cây có giá trị khóa là số chẵn.**

*Trả lời:*















