Ghi chú:

* Làm cá nhân
* Deadline: xem trên moodle

**BÀI TẬP CHƯƠNG DANH SÁCH LIÊN KẾT ĐƠN \_ BUỔI 2**

BÀI TẬP

File trả lời theo mẫu sau:

| **Câu** | **Đáp án** |
| --- | --- |
| Câu 1 |  |
| Câu 2 |  |
| Câu 3 |  |
| Câu 4 |  |
| Câu 5 |  |
| Câu 6 |  |
| Câu 7 |  |
| Câu 8 |  |
| Câu 9 |  |

Tổng: 09 câu.

Câu 1: Chọn câu đúng khi so sánh Singly linked list và array:

A. Array cho phép truy xuất đến các giá trị trong mảng dễ dàng hơn LL

B. Chèn và xóa dễ hơn trong Linked List

C. Truy cập ngẫu nhiên không được phép trong Linked Lists

D. Kích thước của array phải được cho trước, LL thì không cần.

E. A, B, C, D đều đúng

Câu 2: Tạo danh sách liên kết đơn lưu trữ các số nguyên. Các số được lần lượt thêm vào cuối, vào đầu danh sách, hoặc sau một số biết trước. Input gồm nhiều dòng, mỗi dòng sẽ có cấu trúc ở 1 trong 4 dạng sau:

- Dạng 0: Dòng bắt đầu bằng con số 0, theo sau là một số nguyên, chương trình thêm con số này vào đầu danh sách.

- Dạng 1: Dòng bắt đầu bằng con số 1, theo sau là một số nguyên, chương trình thêm con số này vào cuối danh sách.

- Dạng 2: Dòng bắt đầu bằng con số 2, theo sau là 2 số nguyên X và Y, chương trình thêm con số Y vào sau con số X.

Hãy cho biết kết quả danh sách liên kết được in từ đầu đến cuối mảng.

| **STT** | **Input** | **Output** |
| --- | --- | --- |
| 1 | 0 108  0 199  1 465  1 2  0 45  1 90  1 -54  1 -654  1 -313  0 -89  0 1  1 2  1 0  0 -999  1 999 |  |
| 2 | 0 6  2 4 5  0 4  1 2  2 2 7  -1 |  |
| 3 | 0 108  1 199  1 465  0 2  1 45  0 90  2 90 6  0 -654  0 -313  2 90 6  2 50 10  2 15 199  0 0  1 -999  0 999  -1 |  |

Câu 3: Cho danh sách liên kết được quản lý bằng con trỏ đầu và cuối. Hỏi phép toán nào sau đây phụ thuộc vào độ dài của danh sách.

A. Xóa phần tử đầu danh sách

B. Chèn 1 phần tử vào đầu danh sách

C. xóa phần tử cuối của danh sách

D. Thêm 1 giá trị vào cuối danh sách

Câu 4: Khai báo “NODEPTR nodeptr” đúng với trường hợp khai báo nào sau đây. Biết rằng biến nodeptr là biến con trỏ.

| a.  typedef struct node {  int data;  node \*nextPtr;  }\* NODEPTR; | b.  struct node {  int data;  struct node \* nextPtr;  };  typedef struct node \* NODEPTR; |
| --- | --- |
| c.  struct node {  int data;  struct node \* nextPtr;  };  typedef struct node NODEPTR; | d.  typedef struct node {  int data;  NODEPTR nextPtr;  }NODEPTR; |

Câu 5: Cho hàm duyệt và in các giá trị trong danh sách theo thứ tự từ đầu đến cuối mảng như sau. Biết hàm nhận vào giá trị là con trỏ lưu địa chỉ phần tử đầu danh sách.

void traverse(struct Node \*head) {

while (head->next!= NULL) {

cout << head->data;

head = head->next;

}

}

Chọn phát biểu sai:

A. Hàm lỗi khi danh sách rỗng.

B. Hàm không in node cuối khi ds không rỗng.

C. Hàm thực thi không chính xác vì head thay đổi.

Câu 6: DSLK được quản lý bằng con trỏ start, hàm sau sẽ cho ra kết quả ntn nếu ta áp dụng trên danh sách: 1->2->3->4->5->6?

void func(struct node\* start) {

if(start == NULL)

return;

printf("%d ", start->data);

if(start->next != NULL )

func(start->next->next);

printf("%d ", start->data);

}

A. 1 4 6 6 4 1

B. 1 3 5 1 3 5

C. 1 2 3 5

D. 1 3 5 5 3 1

Câu 7: Cho hàm f được viết như sau:

struct item {

int data;

struct item \* next;

};

int f(struct item \*p) {

return (

(p == NULL) ||

(p->next == NULL) ||

((p->data <= p->next->data) && f(p->next))

);

}

Cho sẵn 1 danh sách liên kết p, hỏi hàm f trả về 1 khi và chỉ khi:

A. Tất cả các node trong danh sách có giá trị phân biệt.

B. Danh sách được sắp xếp tăng dần theo giá trị

C. Danh sách được sắp xếp giảm dần theo giá trị

D. A, B, C đều sai

Câu 8: Nếu DSLK được quản lý bằng con trỏ *head*, hàm sau sẽ cho ra kết quả như thế nào?

void func1(struct node\* head) {

if(head == NULL)

return;

func1(head->next);

printf("%d ", head->data);

}

A. In ra tất cả các node từ đầu đến cuối ds

B. In ra tất cả các node theo thứ tự đảo

C. In ra luân phiên các node từ đầu đến cuối ds

D. In ra luân phiên các node theo thứ tự đảo ngược

Câu 9: Cho biết một con trỏ trỏ tới node X của DSLK đơn. Giả sử không cho biết con trỏ head của ds. Hỏi có thể xóa node X từ DSLK không?

A. Có thể nếu X không phải node cuối cùng. Theo 2 bước sau: (1) Copy the data of next of X to X. (2) Delete next of X.

B. Có thể nếu kích thước ds chẵn

C. Có thể nếu kích thước ds lẻ

D. Có thể nếu X không phải node đầu tiên. Theo 2 bước sau: (1) Copy the data of next of X to X. (2) Delete next of X.

Hãy viết hàm tương ứng với yêu cầu trên.

HẾT.