

ÔN TẬP

STT	Nội dung		Lưu ý
1	Vào ra dl	cin, cout	cout: xâu, biến, biểu thức, hàm; cin: biến
2	Rẽ nhánh	if, if...else, switch	btđk nằm trong ngoặc tròn switch: lệnh tương ứng với case thực hiện cho đến khi gặp break hoặc } kết thúc
3	Lặp	for, while, do...while	Kết hợp với break, countious
4	Mảng		Số pt là hằng số, xđ từ khi khai báo. <ul style="list-style-type: none"> - Nhập/xuất mảng - Duyệt mảng (for (int i=0; i<n; i++)) - Max, min (gán =a[0]; for i=1) - Số chính phương - Số nguyên tố - Tìm kiếm, chỉ số (có thể dùng 1 mảng khác) - Sắp xếp - Ktra tính tăng dần/giảm dần (for, break, dựa vào i để kết luận) - Xóa, thêm: tiến lên/lùi xuống, tăng/giảm kích thước mảng
5	Xâu		<ul style="list-style-type: none"> - Có 1 kí tự kết thúc xâu '\0' (while s[i]!='\0', s.size()) - Hàm đơn giản: toupper, isdigit,... - Hàm phức tạp trong xâu: find, substr, erase, insert, append,...

			<ul style="list-style-type: none"> - Bài toán: đảo xuôi, tìm kiếm, in ra chỉ số xuất hiện, tần suất xuất hiện, xóa, thêm,...
6	Con trỏ, cấp phát động		<ul style="list-style-type: none"> - Con trỏ là ĐỊA CHỈ ô nhớ nơi nó trỏ tới - New, delete, delete [],... - Con trỏ, con trỏ mảng (int *p=new int, int *p=new int[n])
7	Struct		<ul style="list-style-type: none"> - Lưu trữ các đối tượng có nhiều thuộc tính với các kiểu dl khác nhau. - Truy cập các thành phần: dấu . - Struct đơn, Mảng struct (A[i], con trỏ struct (p[i]; *(p+i)) - Node *p=new node, Node *p=new node[n]
8	DSLK		<ul style="list-style-type: none"> - Lưu trữ các pt ở các vị trí có thể không liên tục (rời nhau) trong bộ nhớ. - Ngoài trường dl còn có trường con trỏ để trỏ đến pt tiếp theo trong DS - DSLK đơn: tính 1 chiều (từ pt trước đi được đến pt sau nó) - Luôn luôn giữ con trỏ đầu DS, dùng con trỏ khác để chạy duyệt DS - 3 hàm: khởi tạo DS rỗng, tạo nút, tạo DS (thêm đầu, thêm cuối) - Bài toán: giống như mảng

CẤU TRÚC ĐỀ THI

Câu 1: Tính toán, rẽ nhánh, lặp

Câu 2: Mảng

Câu 3: Xâu

Câu 4: Con trỏ, mảng động, DSLK

Mảng	Con trỏ
<pre>for (int i=0; i<n; i++) {xử lý}</pre>	<pre>for (node *p=dau; p!=NULL; p=p->tiep) { xử lý}</pre>
<pre>i = 0; while (i<n) { xử lý i++; }</pre>	<pre>p=dau; while (p!=NULL) { xử lý p=p->tiep; }</pre>