DANH SÁCH LIÊN KẾT ĐÔI

(1) Lý thuyết:

- Xem bài giảng CHƯƠNG 4 (slide 21-33) và giáo trình trang 68-89.

(2) Bài tập tại lớp:

- Viết các chương trình theo các ví dụ trong slide bài giảng và giáo trình chương 4.

```
#define _CRT_SECURE_NO_WARNINGS
#include<stdio.h>
#include<conio.h>
#include<malloc.h>

//Khai báo cấu trúc 1 nút trong danh sách liên kết đôi
struct node
{
	int info; //chứa nội dung của nút
	node *left,*right; //con trỏ chỉ nút trước (bên trái), sau (bên phải) trong d/s
};

//Khai báo kiểu con trỏ chỉ đến nút
typedef node *NODEPTR;

//Cấp phát biến động làm 1 nút cho d/s
NODEPTR GetNode()
{
	NODEPTR p;
	p = (NODEPTR)malloc(sizeof(node));
	return p;
}

//Giải phóng biến động đã cấp phát trước đó
void FreeNode(NODEPTR p)
{
	free(p);
}

//Khởi động dslk  kép
void Initialize(NODEPTR &plist)
{
	plist = NULL;
}

//Kiểm tra d/s có bị rỗng hay không
int Empty(NODEPTR plist)
{
	return (plist == NULL ? 1 : 0);
}
```

```
//Xác định con trỏ của nút thứ i (i = 0,1,2,...) trong dslk kép
NODEPTR NodePointer(NODEPTR plist, int i)
{
	if (i < 0)
		return NULL;
	else
	{
		NODEPTR p = plist;
		int pos = 0;
		while (p != NULL && pos < i)
		{
			p = p->right;
			pos++;
		}
		return p;
	}
}

//Xác định vị trí nút p trong dslk kép
int Position(NODEPTR plist, NODEPTR p)
{
	NODEPTR q = plist;
	int pos = 0;
	while (q != NULL && q != p)
	{
		q = q->right;
		pos++;
	}
	if (q != NULL)
		return pos;//vị trí của nút p trong d/s
	else
		return -1;//nút p không tồn tại
}

//Xác định nút trước của nút p trong dslk kép
NODEPTR PreNode(NODEPTR plist, NODEPTR p)
{
	if (p == plist)
		return NULL;
	else
	{
		NODEPTR q = plist;
		//Di chuyển q đến nút trước nút p
		while (q != NULL && q->right != p)
			q = q->right;
		return q;
	}
}
```

```
//Thêm nút có nội dung x vào đầu dslk
void Push(NODEPTR &plist, int x)
{
	NODEPTR p = GetNode();
	p->info = x;
	if (Empty(plist))//nếu d/s rỗng
	{
		p->left = NULL;
		p->right = NULL;
		plist = p;
	}
	else
	{
		p->right = plist;
		plist->left = p;
		p->left = NULL;
		plist = p;
	}
}

//Thêm một nút có nội dung x ngay sau nút p
void InsRight(NODEPTR p, int x)
{
	if (p == NULL)
		printf("Nut p khong ton tai\n");
	else
	{
		//khai báo con trỏ nút q có nội dung x muốn thêm sau con trỏ nút p
		NODEPTR q = GetNode();
		q->info = x;
		//Khai báo con trỏ r trỏ đến nút sau của p
		NODEPTR r = p->right;
		//Tạo liên kết giữa q và r
		if (p->right == NULL)//nếu p là nút cuối d/s
			q->right = NULL;
		else
		{
			r->left = q;
			q->right = r;
		}
		//Tạo liên kết giữa p và q
		q->left = p;
		p->right = q;
	}
}
```

```
//Thêm một nút có nội dung x ngay trước nút p
void InsLeft(NODEPTR &plist, NODEPTR p, int x)
{
	if (p == NULL)
		printf("Nut p khong ton tai\n");
	else if (p == plist)
		Push(plist,x);//gọi hàm thêm nút có nội dung x vào đầu d/s
	else
	{
		NODEPTR q = GetNode();
		q->info = x;
		p->left->right = q;
		q->left = p->left;
		q->right = p;
		p->left = q;
	}
}

//Xoá nút đầu trong dslk
void Pop(NODEPTR &plist)
{
	NODEPTR p = plist;
	int x = p->info;
	if (plist->right == NULL)//Nếu d/s chỉ có 1 nút
		plist = NULL;//xóa nút đầu
	else
	{
		plist = p->right;
		p->right = NULL;
		plist->left = NULL;
	}
	FreeNode(p);
	printf("\nDa xoa nut dau co gia tri la %d", x);

}

//Xoá nút p trong dslk kép
void DelNode(NODEPTR &plist,NODEPTR p)
{
	if (p == NULL)
		printf("Nut p khong ton tai\n");
	else if (p == plist)//p là nút đầu
		Pop(plist);
	else//p không phải nút đầu
	{
		int x = p->info;
		NODEPTR l = p->left, r = p->right;//l: nút trước p, r: nút sau p
		//Tạo 2 liên kết giữa l và r
		if (r != NULL)
			r->left = l;//tạo liên kết trái cho r
		l->right = r;//tạo liên kết phải cho l
		p->left = NULL;//ngắt liên kết trái của p
		p->right = NULL;//ngắt liên kết phải của p
		FreeNode(p);//xóa nút p
		printf("\nDa xoa nut p co gia tri la %d\n", x);
	}
}
```

```
//Duyệt xuôi danh sách liên kết kép
void RightTraverse(NODEPTR plist)
{
	if (Empty(plist))
		printf("Danh sach bi rong\n");
	else
	{
		NODEPTR p = plist;//khai báo con trỏ p trỏ đến đầu d/s
		while (p != NULL)//lặp trong khi chưa đến nút cuối
		{
			printf("%d\t", p->info);//in giá trị nút hiện thời
			p = p->right;//di chuyển đến nút sau
		}
		printf("\n");
	}
}

//Duyệt ngược danh sách liên kết kép
void LeftTraverse(NODEPTR plist)
{
	if (Empty(plist))
		printf("Danh sach bi rong\n");
	else
	{
		NODEPTR p = plist;
		//Di chuyển đến cuối d/s
		while (p->right != NULL)
			p = p->right;
		//Lần theo liên kết trái, di chuyển từ nút cuối về lại nút đầu
		while (p!=NULL)//lặp trong khi chưa về nút đầu
		{
			printf("%d\t", p->info);//in giá trị nút hiện thời
			p = p->left;//di chuyển về nút trước
		}
		printf("\n");
	}
}

//Tìm nút có nội dung x trong danh sách liên kết kép
NODEPTR Search(NODEPTR plist, int x)
{
	NODEPTR p = plist;
	while (p != NULL && p->info != x)
		p = p->right;
	return p;
}

//Xoá danh sách liên kết
void ClearList(NODEPTR &plist)
{
	while (plist != NULL)//lặp trong khi chưa hết nút
		Pop(plist);//gọi hàm xóa nút đầu
}
```

```
void main()
{
	NODEPTR plist;
	Initialize(plist);
	int chon;
	char tl;
	do
	{
		printf("\n-----CHUONG TRINH XU LY DANH SACH LIEN KET KEP-----\n");
		printf("1. Them 1 nut x vao dau danh sach\n");
		printf("2. Them 1 nut x sau nut p trong danh sach\n");
		printf("3. Them 1 nut x truoc nut p trong danh sach\n");
		printf("4. Xoa nut dau danh sach\n");
		printf("5. Xoa nut p trong danh sach\n");
		printf("6. Duyet xuoi danh sach\n");
		printf("7. Duyet nguoc danh sach\n");
		printf("8. Xoa danh sach\n");
		printf("0. Thoat CT\n");
		printf("-----------------------------------------------------\n");
		printf("Ban chon: ");
		scanf("%d",&chon);
		switch (chon)
		{
		case 0: printf("Dang thoat CT..."); break;
		case 1:
			int x;
			printf("Nhap gia tri nut: ");
			scanf("%d",&x);
			Push(plist, x); //gọi hàm thêm nút x vào đầu dslk
			printf("Danh sach lien ket kep hien tai la:\n");
			RightTraverse(plist); //xuất giá trị các nút trong dslk
			break;
		case 2:
			if (Empty(plist))
				printf("D/s rong\n");
			else
			{
				int vtp;
				printf("Nhap vi tri nut p: ");
				scanf("%d", &vtp);
				//Xác định con trỏ của nút p (ở vị trí vtp) trong dslk
				NODEPTR p = NodePointer(plist, vtp);
				printf("Nhap gia tri nut: ");
				scanf("%d", &x);
				InsRight(p,x);//Gọi hàm thêm nút x sau nút p trong dslk
				printf("Danh sach lien ket kep hien tai la:\n");
				RightTraverse(plist); //xuất giá trị các nút trong dslk
			}
			break;
```

```
case 3:
	if (Empty(plist))
		printf("D/s rong\n");
	else
	{
		int vtp;
		printf("Nhap vi tri nut p: ");
		scanf("%d", &vtp);
		//Xác định con trỏ của nút p (ở vị trí vtp) trong dslk
		NODEPTR p = NodePointer(plist, vtp);
		printf("Nhap gia tri nut: ");
		scanf("%d", &x);
		InsLeft(plist,p, x);//Gọi hàm thêm nút x trước nút p trong dslk
		printf("Danh sach lien ket kep hien tai la:\n");
		RightTraverse(plist); //xuất giá trị các nút trong dslk
	}
	break;
case 4:
	if (Empty(plist))
		printf("D/s rong\n");
	else
	{
		Pop(plist);
		printf("Danh sach lien ket kep hien tai la:\n");
		RightTraverse(plist);//xuất giá trị các nút trong dslk
	}
	break;
case 5:
	if (Empty(plist))
		printf("D/s rong\n");
	else
	{
		int vtp;
		printf("Nhap vi tri nut p: ");
		scanf("%d", &vtp);
		//Xác định con trỏ của nút p (ở vị trí vtp) trong dslk
		NODEPTR p = NodePointer(plist, vtp);
		DelNode(plist,p);
		printf("\nDanh sach lien ket kep hien tai la:\n");
		RightTraverse(plist);//xuất giá trị các nút trong dslk
	}
case 6:
	if (Empty(plist))
		printf("D/s rong\n");
	else
	{
		printf("Danh sach lien ket kep duyet xuoi la:\n");
		RightTraverse(plist);//xuất giá trị các nút trong dslk
	}
	break;
```

```
		case 7:
			if (Empty(plist))
				printf("D/s rong\n");
			else
			{
				printf("Danh sach lien ket kep duyet nguoc la:\n");
				LeftTraverse(plist);//xuất giá trị các nút trong dslk
			}
			break;
		case 8:
			if (Empty(plist))
				printf("D/s rong\n");
			else
			{
				printf("Ban muon xoa d/s SV? (C/K) ");
				tl = _getch();
				if (tl == 'C' || tl == 'c')
				{
					ClearList(plist);//gọi hàm xóa dslk kép
					printf("\nDa xoa d/s SV\n");
				}
			}
			break;
		default:
			printf("Ban chon sai. Moi chon lai.");
		}
	} while (chon != 0);
	_getch();
}
```

(3) Bài tập về nhà:

Viết CT sử dụng dslk đôi để quản lý các SV với các chức năng sau:

a) Xem d/s sinh viên
b) Thêm sinh viên vào d/s
c) Xóa sinh viên trong d/s
d) Cập nhật thông tin sinh viên
e) Sắp xếp d/s sinh viên theo mã SV
f) Tìm kiếm sinh viên theo mã SV
g) Thêm sinh viên vào d/s đã có thứ tự theo mã SV
h) Xóa toàn bộ d/s sinh viên