LAB 7. CÁC KỸ THUẬT XỬ LÝ XÂU KÝ TỰ

THỜI LƯỢNG: 6 TIẾT

A. Mục tiêu

- Giúp sinh viên hiểu rõ và thực hiện thuần thục các kỹ thuật xử lý xâu ký tự (chuỗi).
- Tiếp tục rèn luyện kỹ năng phát triển chương trình từng bước theo chức năng..
- Sau khi hoàn thành bài thực hành này, sinh viên :
  - Nắm vững các khái niệm, định nghĩa và thao tác nhập, xuất xâu ký tự.
  - Nắm vững các kỹ thuật xử lý cơ bản trên xâu ký tự.
  - Biết cách định nghĩa và sử dụng kiểu dữ liệu xâu ký tự bằng từ khóa **typedef**.
  - Phân biệt được mảng 1 chiều và xâu ký tự.
  - Truyền tham số thực, truyền tham biến.

B. Yêu cầu

- Kết quả thực tập phần D (hướng dẫn thực hành) được thực hiện tại phòng Lab theo yêu cầu :
  - Tạo thư mục, đặt tên là **MSSV_Lab07_D_HD,** để lưu bài làm. Trong đó, MSSV là mã số của sinh viên.
  - Các bài 1,2: tạo các project theo hướng dẫn, lưu trữ trong thu mục trên
  - Xóa thư mục Debug trong các project
  - Nén thư mục **MSSV_Lab07_D_HD**
  - Thời gian thực hiện : 2 tiết
  - ***Giáo viên thu bài qua mạng tại phòng lab vào cuối buổi thực tập thứ 11***
- Kết quả thực tập phần E (bài tập bắt buộc) được thực hiện tại phòng Lab theo yêu cầu :
  - Tạo thư mục, đặt tên là **MSSV_Lab07_E_BB,** để lưu bài làm.
  - Các bài 1, 2, 3: tạo các project theo hướng dẫn, lưu trữ trong thu mục trên
  - Xóa thư mục Debug trong các project
  - Nén thư mục **MSSV_Lab0E_E_BB**
  - Thời gian thực hiện : 4 tiết
  - ***Giáo viên thu bài qua mạng tại phòng lab vào cuối buổi thực tập thứ 12***

C. Ôn tập lý thuyết

1. Cú pháp khai báo biến chuỗi ký tự

Cú pháp: **char** **Tên_biến_chuỗi [Số_ký_tự_tối_đa];**
Hoặc: **wchar_t** **Tên_biến_chuỗi [Số_ký_tự_tối_đa];**
Trong đó:

- **Tên_biến_chuỗi**: là tên biến chuỗi ký tự, do người lập trình đặt và tuân theo quy tắc đặt tên.
- **Số_ ký_tự_tối_đa** là một nguyên dương, cho biết số ký tự tối đa có thể lưu trong biến chuỗi.

2. Cú pháp định nghĩa kiểu dữ liệu chuỗi ký tự

Cú pháp: **typedef char** **Tên_kiểu_chuỗi [Số_ký_tự_tối_đa];**

Ví dụ: **#define MAX** **100**

```
typedef  char   Chuoi [MAX];
```

Khi đó:

- **Chuoi**: trở thành 1 kiểu dữ liệu, là kiểu chuỗi ký tự.
- Ta có thể khai báo các biến thuộc kiểu này: **Chuoi a, b**.

3. Các thao tác nhập - xuất chuỗi ký tự

a. Nhập xâu ký tự từ bàn phím

Với khai báo xâu ký tự: **char a**[MAX]; hoặc **Chuoi a**;

Ta sử dụng 1 trong 2 cách sau để nhập dữ liệu cho **a** :

- cin >> a; Cách này không thể nhập chuỗi có chứa khoảng trắng.
- gets_s(a); **a** có chứa khoảng trắng, phải dùng hàm _flushall(); trước khi gọi hàm gets_s().

Lưu ý quan trọng:

- Một xâu ký tự bao giờ cũng phải kết thúc bởi ký tự NULL (\0).
- Đối với các cách trên, khi kết thúc việc nhập (nhấn Enter), trình biên dịch sẽ tự động thêm ký tự NULL vào cuối chuỗi.
- Nếu ta dùng cách nhập từng ký tự như mảng một chiều thông thường (bằng 1 vòng lặp for) thì khi đó, **a** chỉ là mảng một chiều có chứa ***n*** các ký tự khác NULL. Để **a** biến thành chuỗi ký tự chứa ***n*** ký tự khác NULL, cần thêm ký tự NULL vào cuối mảng, tức là:

```
for (int i=0, i<n; i++)
        cin >> a[i];
a[n] = NULL;
```

- Ký tự đầu tiên của chuỗi **a** bao giờ cũng tương ứng với chỉ số 0, nghĩa là: **a**[0] là ký tự đầu tiên trong chuỗi. Nếu **a** là chuỗi rỗng thì **a**[0] = NULL.
- Nếu chuỗi **a** có chứa ***n*** ký tự khác NULL (hay chiều dài là ***n***) thì **a**[***n***] = NULL.

b. Xuất giá trị của chuỗi ra màn hình

Dùng lệnh cout như khi xuất các biến khác: **cout << a**;

4. Duyệt các ký tự trong chuỗi

Sử dụng tín hiệu kết thúc xâu ký tự là NULL, ta thường dùng một trong hai cách sau để duyệt:

```
for (int i=0; a[i] != NULL; i++)
        // Xử lý a[i] ở đây
```

```
int i=0;
while (a[i] != NULL)
        {
                // Xử lý a[i] ở đây
                i++;
        }
```

5. Một số hàm trong thư viện <string.h> :

Tên hàm	Chức năng	Nguyên mẫu	Sử dụng
gets_s	Nhập dữ liệu cho chuỗi (không quá MAX ký tự)	char * __cdecl **gets_s**(char * _Buf, _In_ rsize_t _Size)	gets_s(a,MAX);
flushall()	Làm (xóa) trống vùng đệm bàn phím	int __cdecl **_flushall()**	_flushall(); int t = _flushall();
strlen	Tính chiều dài chuỗi	size_t **strlen**(const char *a);	int l = strlen(a);

strcat_s	Nối chuỗi (nối chuỗi sau vào cuối chuỗi trước)	errno_t __cdecl **strcat_s**(char * _Dst, _In_ rsize_t _SizeInBytes, _In_z_ const char * _Src)	Strcat_s(a,MAX,b);
strcpy_s	Sao chép chuỗi (chép chuỗi sau sang chuỗi trước với không quá MAX ký tự)	errno_t **strcpy_s**(char *strDestination, size_t numberOfElements, const char *strSource);	strcpy_s(a,MAX,b)
strcmp	So sánh 2 chuỗi (có phân biệt ký tự thường, hoa)	int **strcmp**(const char *s1, const char *s2);	$strcmp(s1,s2)=q\begin{cases}<0; s1<s2\\=0; s1\equiv s2;\\>0; s1>s2\end{cases}$
_strcmpi	So sánh 2 chuỗi (có phân biệt ký tự thường, hoa)	int **_strcmpi**(const char *s1, const char *s2);	$\_strcmpi(s1,s2)=q\begin{cases}<0; s1<s2\\=0; s1\equiv s2;\\>0; s1>s2\end{cases}$
. . .			

D. Hướng dẫn thực hành

Phần này xây dựng chương trình thực hiện các thao tác trên cấu trúc dữ liệu xâu ký tự với các chức năng cơ bản.
Tiếp tục tổ chức chương trình theo thư viện và có/không có hệ thống menu.

Bài 1:

Viết chương trình thực hiện các thao tác trên các chuỗi (xâu ký tự) , với các chức năng cơ bản :

0. ***Thoát khỏi chương trình***
1. ***Nhập chuỗi***
2. ***Xem chuỗi***
3. ***Tính chiều dài chuỗi***
4. ***Nối chuỗi***
5. ***Sao chép chuỗi***
6. ***So sánh chuỗi theo thứ tự từ điển (không phân biệt ký tự thường, HOA)***
7. ***So sánh chuỗi theo thứ tự từ điển (có phân biệt ký tự thường, HOA)***

(Các chức năng từ 3 đến 7, vừa dùng các hàm thư viện trong string.h, vừa tự cài đặt).

Bước 1. Tạo dự án Win32 Console Application mới. Đặt tên là ***Lab07_D_Bai1***

Bước 2. Tạo cấu trúc cho chương trình như đã hướng dẫn trong **mục 2 lab 4** (từ 1- 8 để có cấu trúc nội dung tối thiểu chạy được chương trình).

Tức là ta có kết quả chương trình ở bước này như sau :

- Tập tin ***thuvien.h*** : Rỗng
- Tập tin ***menu.h*** : Rỗng
- Tập tin ***program.cpp*** có nội dung như sau :

```
#include <iostream>
#include <conio.h>
using namespace std;
```

```
#include "thuvien.h"
#include "menu.h"

void ChayChuongTrinh();

int main()
{
	ChayChuongTrinh();
	return 1;
}
void ChayChuongTrinh()
{
	_getch();
}
```

Nhấn Ctrl + F5 để chạy chương trình, sửa lỗi nếu có.

Bước 3:

Bước này ta định nghia kiểu dữ liệu mới; tổ chức và vận hành hệ thống menu.

- **Trong tập tin *thuvien.h* :**

Ta bổ sung định nghia hằng, kiểu dũ liệu mới :

Vì chương trình thực hiện các thao tác trên xâu ký tự, nên ta cần định nghĩa một hằng là giá trị kích thước khai báo của xâu. Ngoài ra, ta có thể định nghĩa một kiểu dữ liệu xâu ký tự (lưu ý rằng có thể không cần thực hiện định nghĩa này vì ta có thể làm trực tiếp trên biến xâu ký tự)

```
//Dinh nghia hang
#define MAX  100//kich thuoc khai bao xau ky tu

//Dinh nghia kieu du lieu moi:
typedef char String[MAX]; //kiểu xau ký tự đặt tên String

//Khai bao nguyen mau cac ham xu ly, nhap xuat
	//bổ sung sau

//Dinh nghia cac ham xu ly, nhap xuat
	//bổ sung sau
```

- **Trong tập tin *menu.h* :**

```
// Khai báo nguyên mẫu các hàm xử lý menu
	//bổ sung sau
 // Định nghĩa các hàm xử lý menu

3.1 Định nghĩa hàm xuất danh sách chức năng ra màn hình

void XuatMenu()
{
	cout << "\n================== Bang Menu ==================";
	cout << "\n0. Thoat khoi chuong trinh";
	cout << "\n1. gets_s : Nhap chuoi";
	cout << "\n2. Xem chuoi";
```

```
	cout << "\n3. strlen_Tinh chieu dai chuoi";
	cout << "\n4. strcat_s_Noi chuoi sau vao sau chuoi truoc";
	cout << "\n5. strcpy_s_Chep chuoi sau vao chuoi truoc";
	cout << "\n6. _strcmpi_So sanh chuoi _ khong phan biet KT thuong, HOA";
	cout << "\n7. strcmp_So sanh chuoi _ phan biet KT thuong, HOA";
	//=====================================
	cout << "\n8. Noi chuoi sau vao sau chuoi truoc";
	cout << "\n9. Chep chuoi sau qua chuoi truoc";
	cout << "\n10. So sanh chuoi _ khong phan biet KT thuong, HOA";
	cout << "\n11. So sanh chuoi _ phan biet KT thuong, HOA";
}
```

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.

3.2 Định nghĩa hàm chọn một menu trong danh sách

```
// Input : soMenu = Số lượng menu có thể chọn.
// Output: Số thứ tự menu do người dùng nhập vào.
int ChonMenu(int soMenu)
{
	int stt;
	for (;;)
	{
		system("CLS");
		XuatMenu();
		cout<<"\nNhap 1 so khong khoang [0,.." << soMenu << "] de chon chuc nang, stt = ";
		cin >> stt;
		if (0 <= stt && stt <= soMenu)
			break;
	}
	return stt;

}
```

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.

3.3 Định nghĩa hàm xử lý menu :

Các thao tác thường thực hiện trên cùng một đầu vào là 2 chuỗi, nên ta bổ sung thêm 2 biến chuỗi kiểu String làm đối của hàm **XuLyMenu**, ngoài tham số đã có là tham số menu.

```
void XuLyMenu(int menu, String a, String b)
{
	//khai bao bien

	switch (menu)
	{
		case 0:
			system("CLS");
			cout << "\n0. Thoat khoi chuong trinh\n";
			break;
		case 1:
			system("CLS");
			cout << "\n1. gets_s_Nhap chuoi a: ";
			break;

		case 2:
			system("CLS");
```

```
                cout << "\n2. Xem chuoi";
                break;


        case 3:
                system("CLS");
                cout << "\n3. strlen_Tinh chieu dai chuoi";
                break;


        case 4:
                system("CLS");
                cout << "\n4. strcat_s_Noi chuoi sau vao sau chuoi truoc";
                break;

        case 5:
                system("CLS");
                cout << "\n5. strcpy_s_Chep chuoi sau qua chuoi truoc";
                break;

        case 6:
                system("CLS");
                cout << "\n6. _strcmpi _ So sanh chuoi _khong phan biet KT thuong, HOA";
                break;

        case 7:
                system("CLS");
                cout << "\n7. strcmp _ So sanh chuoi _ phan biet KT thuong, HOA";
        //============================

        case 8:
                system("CLS");
                cout << "\n8.Noi chuoi sau vao sau chuoi truoc";
                break;

        case 9:
                system("CLS");
                cout << "\n9. Chep chuoi sau qua chuoi truoc";
                break;

        case 10:
                system("CLS");
                cout << "\n10. So sanh chuoi _khong phan biet KT thuong, HOA";
                break;

        case 11:
                system("CLS");
                cout << "\n7. So sanh chuoi _ phan biet KT thuong, HOA";
                break;

    }
    _getch();
}
```

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.

3.4 Bổ sung khai báo nguyên mẫu các hàm tổ chức menu trong phần khai báo nguyên mẫu hàm

```
void XuatMenu();
int ChonMenu(int soMenu);
void XuLyMenu(int menu, String a, String b);
```

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có

- Trong tập tin ***program.cpp*** :

  + Hàm ***ChayChuongTrinh*** ta cập nhật lại như sau :

```
void ChayChuongTrinh()
{
	int menu,
		soMenu = 11;
	String a,b;
	do
	{
		system("CLS");
		menu = ChonMenu(soMenu);
		XuLyMenu(menu, a,b);
	} while (menu > 0);

}
```

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.

Kiểm tra sự vận hành của hệ thống menu.

Kiểm tra chức năng thoát khỏi chương trình (chọn 0).

Bước 4 : Bổ sung vào chương trình các thao tác nhập xuất dữ liệu

Trong bước 4 này, ta làm công việc sau :

- Trong ***program.cpp*** Bổ sung thêm thư viện <***string.h***>
- Trong tập tin ***thuvien.h*** , soạn thảo các hàm nhập, xuất chuỗi
- Trong tập tin ***menu.h*** bổ sung xử lý chức năng nhập chuỗi,xem chuỗi trong hàm ***XuLyMenu.***

- Trong tập tin *thuvien.h* :

Bổ sung các hàm nhập xuất :

4.1 Hàm nhập chuỗi

(Sử dụng hàm gets_s trrong string. h)

```
void gets_s_NhapChuoi(String a, char kt)
{
	cout << "\nNhap chuoi: "<<kt<<"  = ";
	_flushall();
	gets_s(a,MAX);
}
```

Ghi chú : Có thể không cần viết hàm, mà dùng trực tiếp hàm gets_s

4.2 Hàm xuất dữ liệu ma trận vuông ra màn hình

```
//Xuat chuoi
void XuatChuoi(String a)
{
	cout << a;
}
```

Ghi chú : có thể dùng trực tiếp cout<<.

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.

4.3 Bổ sung nguyên mẫu các hàm :

```
void gets_s_NhapChuoi(String a, char kt);
void XuatChuoi(String a);
```

- Trong tập tin *menu.h* :

Bổ sung xử lý chức năng nhập chuỗi vào case 1, xem chuỗi vào case 2 (Các case từ 3 đến 11 giữ nguyên)

```
void XuLyMenu(int menu, String a, String b)
{
	//khai bao bien
	int kq;
	switch (menu)
	{
		case 0:
			system("CLS");
			cout << "\n0. Thoat khoi chuong trinh\n";
			break;
		case 1:
			system("CLS");
			cout << "\n1. gets_s_Nhap chuoi a: ";
			gets_s_NhapChuoi(a, 'a');
			cout << "\nChuoi a vua nhap: ";
			XuatChuoi(a);

			NhapChuoi(b,'b');
			cout << "\nChuoi b vua nhap: ";
			XuatChuoi(b);

			cout << "\nNhan phim bat ky de tiep tuc";

			break;

		case 2:
			system("CLS");
			cout << "\n2. Xem chuoi";
			cout << "\nChuoi a: ";
			XuatChuoi(a);
			cout << "\nChuoi b: ";
			XuatChuoi(b);
			cout << "\nNhan phim bat ky de tiep tuc";

			break;

		//…

	}

	_getch();

}
```

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.
Kiểm tra kết quả thực hiện các chức năng 1, 2.

Trong các bước tiếp theo, ta bổ sung lần lượt từng chức năng vào chương trình :

- Lần lượt soạn thảo từng hàm chức năng trong tập tin ***thuvien.h***,
- Lần lượt bổ sung xử lý chức năng trong hàm ***XuLyMenu*** của ***menu.h***,

Bước 5 : Bổ sung chức năng nối chuỗi vào chương trình

- Trong tập tin *thuvien.h* :

5.1 Hàm tính chiều dài chuỗi

(Sử dụng hàm strlen trong string. h)

```
int strlen_TinhChieuDaiChuoi(String a)
{
	int l = strlen(a);
	return l;
}
```

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.

5.2 Bổ sung nguyên mẫu các hàm :

```
int strlen_TinhChieuDaiChuoi(String a);
```

- Trong tập tin *menu.h* :

Trong hàm XuLyMenu khai báo thêm biến kq kiểu int để lưu trữ kết quả tính toán, bổ sung xử lý chức năng tính chiều dài chuỗi vào case 3 (Các case khác giữ nguyên)

```
void XuLyMenu(int menu, String a, String b)
{
	//khai bao bien
	int kq;
	switch (menu)
	{
		//. . .
		case 3:
			system("CLS");
			cout << "\n3. strlen_Tinh chieu dai chuoi";
			cout << "\nChuoi a: ";
			XuatChuoi(a);
			cout << "\nChieu dai chuoi a: l = " << strlen_TinhChieuDaiChuoi(a);
			cout << "\nChuoi b: ";
			XuatChuoi(b);
			cout << "\nChieu dai chuoi b: l = " << strlen_TinhChieuDaiChuoi(b);
			cout << "\nNhan phim bat ky de tiep tuc";
			break;
		//…

	}

	_getch();

}
```

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.
Kiểm tra kết quả thực hiện các chức năng 3.

Bước 6 :

Bổ sung chức năng nối chuỗi vào chương trình

- Trong tập tin *thuvien.h* :

6.1 Hàm nối chuỗi

(Sử dụng hàm strcat_s trong string. h, nối chuỗi nguồn b vào cuối chuỗi đích a)

```
void strcat_s_Noi_ChuoiSau_VaoSau_ChuoiTruoc(String a, String b)
{
	strcat_s(a, MAX, b);
}
```

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.

6.2 Bổ sung nguyên mẫu các hàm :

```
void strcat_s_Noi_ChuoiSau_VaoSau_ChuoiTruoc(String a, String b);
```

- Trong tập tin *menu.h* :

Bổ sung xử lý chức năng nối chuỗi vào case 4 (Các case khác giữ nguyên)

```
void XuLyMenu(int menu, String a, String b)
{
	//khai bao bien
	int kq;
	switch (menu)
	{
		//. . .
		case 4:
			system("CLS");
			cout << "\n4. strcat_s_Noi chuoi sau vao sau chuoi truoc";
			cout << "\nChuoi truoc : a = ";
			XuatChuoi(a);
			cout << "\nChieu sau :  b = ";
			XuatChuoi(b);

			strcat_s_Noi_ChuoiSau_VaoSau_ChuoiTruoc(a, b);
			cout << "\nChuoi truoc sau khi noi : a = ";
			XuatChuoi(a);

			break;
		//…

	}

	_getch();

}
```

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.
Kiểm tra kết quả thực hiện các chức năng 4.

Bước 7 :

Bổ sung chức năng sao chép chuỗi vào chương trình

- Trong tập tin *thuvien.h* :

7.1 Hàm sao chép chuỗi

(Sử dụng hàm strcpy_s trong string. h)

```
void strcpy_s_Chep_ChuoiSau_Qua_ChuoiTruoc(String a, String b)
{
	strcpy_s(a, MAX, b);
}
```

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.

7.2 Bổ sung nguyên mẫu các hàm :

```
void strcpy_s_Chep_ChuoiSau_Qua_ChuoiTruoc(String a, String b);
```

- Trong tập tin *menu.h* :

Bổ sung xử lý chức năng sao chép chuỗi vào case 5 (Các case khác giữ nguyên)

```
void XuLyMenu(int menu, String a, String b)
{
	//khai bao bien
	int kq;
	switch (menu)
	{
		//. . .
		case 5:
			system("CLS");
			cout << "\n5. strcpy_s_Chep chuoi sau qua chuoi truoc";
			cout << "\nChuoi sau :  b = ";
			XuatChuoi(b);

			strcpy_s_Chep_ChuoiSau_Qua_ChuoiTruoc(a, b);
			cout << "\nChuoi truoc a,do b chep qua : a = ";
			XuatChuoi(a);
			break;
		//…

	}

	_getch();

}
```

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.
Kiểm tra kết quả thực hiện các chức năng 5.

Bước 8 :

Bổ sung chức năng so sánh chuỗi không phân biệt ký tự thường hoa vào chương trình

- Trong tập tin *thuvien.h* :

8.1 Hàm so sánh chuỗi không phân biệt ký tự thường hoa

(Sử dụng hàm _strcmpi trong string. h)

```
//So sanh 2 chuoi theo thu tu tu dien : khong phan biet thuong hoa
```

```
int _strcmpi_SoSanhChuoi_KPB(String a, String b)
{
	return _strcmpi(a, b);
}
```

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.

8.2 Bổ sung nguyên mẫu các hàm :

```
int _strcmpi_SoSanhChuoi_KPB(String a, String b);
```

- Trong tập tin *menu.h* :

Bổ sung xử lý chức năng so sánh chuỗi chuỗi vào case 6 (Các case khác giữ nguyên)

```
void XuLyMenu(int menu, String a, String b)
{
	//khai bao bien
	int kq;
	switch (menu)
	{
		//. . .
		case 6:
			system("CLS");
			cout << "\n6. strcmpi _ So sanh chuoi _khong phan biet KT thuong, HOA";
			cout << "\nChuoi a = ";
			XuatChuoi(a);
			cout << "\nChieu b = ";
			XuatChuoi(b);

			kq = _strcmpi_SoSanhChuoi_KPB(a, b);
			if (kq == 1)
				cout << "\na > b";
			else
				if(kq == -1)
					cout << "\na < b";
				else
					cout << "\na == b";
			break;

		//…

	}

	_getch();

}
```

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.
Kiểm tra kết quả thực hiện các chức năng 6.

Bước 9 :

Bổ sung chức năng so sánh chuỗi có phân biệt ký tự thường hoa vào chương trình

- Trong tập tin *thuvien.h* :

9.1 Hàm so sánh chuỗi có phân biệt ký tự thường hoa

(Sử dụng hàm strcmp trong string. h)

```
//So sanh 2 chuoi theo thu tu tu dien : co phan biet thuong hoa
int strcmp_SoSanhChuoi_PB(String a, String b)
{
	return strcmp(a, b);
}
```

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.

9.2 Bổ sung nguyên mẫu các hàm :

```
int strcmp_SoSanhChuoi_PB(String a, String b);
```

- Trong tập tin *menu.h* :

Bổ sung xử lý chức năng so sánh chuỗi chuỗi vào case 6 (Các case khác giữ nguyên)

```
void XuLyMenu(int menu, String a, String b)
{
	//khai bao bien
	int kq;
	switch (menu)
	{
		//. . .
		case 7:
			system("CLS");
			cout << "\n7. strcmp _ So sanh chuoi _ phan biet KT thuong, HOA";
			cout << "\nChuoi a = ";
			XuatChuoi(a);
			cout << "\nChieu b = ";
			XuatChuoi(b);

			kq = strcmp_SoSanhChuoi_PB(a, b);
			if (kq == 1)
				cout << "\na > b";
			else
			if (kq == -1)
				cout << "\na < b";
			else
				cout << "\na == b";
			break;
		//…

	}

	_getch();

}
```

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.

Kiểm tra kết quả thực hiện các chức năng 7.

Bước 10 :

Bổ sung chức năng nối chuỗi (tự viết, không dùng hàm thư viện) vào chương trình

- Trong tập tin *thuvien.h* :

10.1 Hàm nối chuỗi

```
//ham noi chuoi b vao cuoi chuoi a
void Noi_ChuoiSau_VaoSau_ChuoiTruoc(String a, String b)
{
	int i, l;
	l = strlen_TinhChieuDaiChuoi(a);
	for (i = 0; b[i] != NULL; i++)
		a[l++] = b[i];
	a[l] = NULL;
}
```

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.

10.2 Bổ sung nguyên mẫu các hàm :

```
void  Noi_ChuoiSau_VaoSau_ChuoiTruoc(String a, String b);
```

- Trong tập tin *menu.h* :

Bổ sung xử lý chức năng nối chuỗi vào case 8 (Các case khác giữ nguyên)

```
void XuLyMenu(int menu, String a, String b)
{
	//khai bao bien
	int kq;
	switch (menu)
	{
		//. . .
		case 8:
			system("CLS");
			cout << "\n4. strcat_s_Noi chuoi sau vao sau chuoi truoc";
			cout << "\nChuoi truoc : a = ";
			XuatChuoi(a);
			cout << "\nChieu sau :  b = ";
			XuatChuoi(b);

			Noi_ChuoiSau_VaoSau_ChuoiTruoc(a, b);
			cout << "\nChuoi truoc sau khi noi : a = ";
			XuatChuoi(a);

			break;
		//…

	}

	_getch();

}
```

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.
Kiểm tra kết quả thực hiện các chức năng 8.

Bước 11 :

Bổ sung chức năng sao chép chuỗi (không dùng hàm thư viện) vào chương trình

- Trong tập tin *thuvien.h* :

11.1 Hàm sao chép chuỗi

```
//Ham sao chep chuoi b sang a
void Chep_ChuoiSau_Qua_ChuoiTruoc(String a, String b)
{
	int i;
	for (i = 0; (a[i] = b[i]) != NULL; i++);
}
```

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.

11.2 Bổ sung nguyên mẫu các hàm :

```
void  Chep_ChuoiSau_Qua_ChuoiTruoc(String a, String b);
```

- Trong tập tin *menu.h* :

Bổ sung xử lý chức năng sao chép chuỗi vào case 9 (Các case khác giữ nguyên)

```
void XuLyMenu(int menu, String a, String b)
{
	//khai bao bien
	int kq;
	switch (menu)
	{
		//. . .

		case 9:
			system("CLS");
			cout << "\n9. Chep chuoi sau qua chuoi truoc";
			cout << "\nChuoi sau :  b = ";
			XuatChuoi(b);

			Chep_ChuoiSau_Qua_ChuoiTruoc(a, b);
			cout << "\nChuoi truoc a,do b chep qua : a = ";
			XuatChuoi(a);

			break;

		//…

	}

	_getch();

}
```

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.
Kiểm tra kết quả thực hiện các chức năng 9.

Bước 12 :

Bổ sung chức năng so sánh chuỗi không phân biệt ký tự thường hoa vào chương trình

- Trong tập tin *thuvien.h* :

12.1 Hàm so sánh chuỗi không phân biệt ký tự thường hoa

```
//So sanh 2 chuoi theo thu tu tu dien : khong phan biet thuong hoa
int SoSanhChuoi_KPB(String a, String b)
{
	int i;
	for (i = 0; a[i] != NULL && b[i] != NULL; i++)
	{
		if (Chuyen_KT_Hoa(a[i]) < Chuyen_KT_Hoa(b[i]))
			return -1;
		if (Chuyen_KT_Hoa(a[i]) > Chuyen_KT_Hoa(b[i]))
			return 1;
	}
	if (a[i] != NULL)
		return 1;
	if (b[i] != NULL)
		return -1;
	return 0;

}
```

12.2 Hàm chuyển ký tự thường thành hoa

```
char Chuyen_KT_Hoa(char x)
{
	if ('a' <= x && x <= 'z')
		x = x - 32;
	return x;
}
```

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.

12.3 Bổ sung nguyên mẫu các hàm :

```
int SoSanhChuoi_KPB(String a, String b);
char Chuyen_KT_Hoa(char x);
```

- Trong tập tin *menu.h* :

Bổ sung xử lý chức năng so sánh chuỗi case 10 (Các case khác giữ nguyên)

```
void XuLyMenu(int menu, String a, String b)
{
	//khai bao bien
	int kq;
	switch (menu)
	{
		//. . .

		case 10:
			system("CLS");
			cout << "\n10. So sanh chuoi _khong phan biet KT thuong, HOA";
			cout << "\nChuoi a = ";
			XuatChuoi(a);
			cout << "\nChieu b = ";
			XuatChuoi(b);
```

```
                        kq = SoSanhChuoi_KPB(a, b);
                        if (kq == 1)
                                cout << "\na > b";
                        else
                                if(kq == -1)
                                        cout << "\na < b";
                                else
                                        cout << "\na == b";
                        break;

                //…

        }

        _getch();

}
```

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.
Kiểm tra kết quả thực hiện các chức năng 10.

Bước 13 :

Bổ sung chức năng so sánh chuỗi có phân biệt ký tự thường hoa vào chương trình

- Trong tập tin *thuvien.h* :

13.1 Hàm so sánh chuỗi có phân biệt ký tự thường hoa

```
//So sanh 2 chuoi theo thu tu tu dien : co phan biet thuong hoa
int SoSanhChuoi_PB(String a, String b)
{
        int i;
        for (i = 0; a[i] != NULL && b[i] != NULL; i++)
        {
                if (a[i] < b[i])
                        return -1;
                if (a[i] > b[i])
                        return 1;
        }
        if (a[i] != NULL)
                return 1;
        if (b[i] != NULL)
                return -1;
        return 0;

}
```

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.

13.2 Bổ sung nguyên mẫu các hàm :

```
int SoSanhChuoi_PB(String a, String b);
```

- Trong tập tin *menu.h* :

Bổ sung xử lý chức năng so sánh chuỗi chuỗi vào case 11 (Các case khác giữ nguyên)

```
void XuLyMenu(int menu, String a, String b)
{
	//khai bao bien
	int kq;
	switch (menu)
	{
		//. . .
		case 11:
			system("CLS");
			cout << "\n11. So sanh chuoi _ phan biet KT thuong, HOA";
			cout << "\nChuoi a = ";
			XuatChuoi(a);
			cout << "\nChieu b = ";
			XuatChuoi(b);

			kq = SoSanhChuoi_PB(a, b);
			if (kq == 1)
				cout << "\na > b";
			else
			if (kq == -1)
				cout << "\na < b";
			else
				cout << "\na == b";
			break;
		//…

	}

	_getch();

}
```

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.
Kiểm tra kết quả thực hiện các chức năng 11.
Kiểm tra các chức năng của chương trình – Kết thúc chương trình.

Ghi chú :

Thử thay thế hàm nhập chuỗi - void gets_s_NhapChuoi(String a, char kt) – bằng hàm sau đây :

```
//Nhap chuoi, không dùng gets_s
void NhapChuoi(String a, char kt)
{
	int i = 0;
	char x;
	cout << "\nNhap Chuoi: " << kt << "  : ";
	_flushall();
	x = _getch();
	while (x != 13)
	{
		a[i++] = x;
		x = _getch();
	}
	a[i] = NULL;
}
```

Bài 2: (nắn tên)

Từ là một dãy các ký tự liên tiếp không chứa dấu cách (ký tự trắng). Ta xem tên là một xâu ký tự gồm nhiều từ phân cách nhau bởi các dấu cách.

Viết chương trình nắn các tên được nhập từ bàn phím theo quy cách:

- Quy cách 1 : Xóa các ký tự trắng ở đầu và cuối tên.
- Quy cách 2 : Khử bớt các ký tự trắng ở giữa hai tên, chỉ giữ lại một ký tự trắng.
- Quy cách 3 : Các chữ cái đầu mỗi từ viết IN HOA, các chữ còn lại viết in thường.

Ví dụ: (hang trên : tên nhập; hang sau : tên đã nắn theo chuẩn

Da LAT hoaNG Hon
Da Lat Hoang Hon

Ý tưởng giải quyết bài toán :

- Xử lý tên -> xử lý từ -> xử lý ký tự
- Trong quả trình xử lý tên theo quy cách, ta dùng một biến trung gian để lưu trử kết quả
- Xử lý tên : Khử các ký tự trắng ở đầu và cuối tên. Khử các ký tự trắng giữa 2 từ chỉ để lại một.
- Xử lý từ : Ký tự đầu nắn thành ký hoa, các ký tự còn lại trong từ nắn thành ký tự thường. Chèn ký tự vào cuối biến trung gian. . .

Nên ta cần thực hiện các công việc sau :

- Chuyển ký tự thường thành hoa.
- Chuyển ký tự hoa thành thường.
- Chèn một ký tự vào cuối xâu ký tự
- Nắn tên theo quy cách
- . . .

Chương trình tổ chức theo thư viện hàm ***(và không có hệ thống menu.)***

Bước 1. Tạo Project rỗng mới đặt tên ***Lab07_D_Bai2***

Bước 2. Tạo cấu trúc cho chương trình như đã hướng dẫn trong **mục 1 lab 4** (từ 1-7 để có cấu trúc nội dung tối thiểu chạy được chương trình, dạng không có hệ thống menu).

Tức là ta có kết quả chương trình ở bước này như sau :

- Tập tin ***thuvien.h*** : Rỗng (chỉ có các chú thích về cấu trúc văn bản)
- Tập tin ***program.cpp*** có nội dung như sau :

```
#include <iostream>
#include <conio.h>
using namespace std;
#include "thuvien.h"
void ChayChuongTrinh();
int main()
{
	ChayChuongTrinh();
	return 1;
}
void ChayChuongTrinh()
{
	_getch();
}
```

Nhấn Ctrl + F5 để chạy chương trình, sửa lỗi nếu có.

Bước 3:

Bước này ta bổ sung thêm các thư viện cần thiết, định nghia hằng, kiểu dữ liệu mới, định nghĩa các hàm chức năng …

- Trong tập tin *program.cpp* :

Bổ sung thêm thư viện ***<string.h>***

- Trong tập tin *thuvien.h* :

+ Bổ sung định nghia hằng, kiểu dữ liệu mới :

Vì chương trình thực hiện các thao tác trên xâu ký tự, nên ta cần định nghĩa một hằng là giá trị kích thước khai báo của xâu.

//Dinh nghia hang

```
#define MAX  100//kich thuoc khai bao xau ky tu
#define CACH   ' ' //ky tu trang
```

//Dinh nghia kieu du lieu moi: không có

//Khai bao nguyen mau cac ham xu ly, nhap xuat

```
//bổ sung sau
```

//Dinh nghia cac ham xu ly, nhap xuat : không (dùng hàm thư viện trong string.h

3.1 Định nghĩa hàm chuyển sang ký tự thường

(Nếu là ký tự thường thì giữ nguyên, nếu là HOA thì chuyển sang thường)

```
char Chuyen_KT_Thuong(char x)
{
	if ('A' <= x && x <= 'Z')
		x = x + 32;
	return x;
}
```

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.

3.2 Định nghĩa hàm chuyển sang ký tự HOA

(Nếu là ký tự HOA thì giữ nguyên, nếu là thường thì chuyển sang HOA

```
char Chuyen_KT_Hoa(char x)
{
	if ('a' <= x && x <= 'z')
		x = x - 32;
	return x;
}
```

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.

3.3 Chèn 1 ký tự vào cuối một xâu ký tự

```
//Chen kt vao cuoi b
void ChenCuoi(char b[MAX], char kt)
{
	int i;
	for (i = 0; b[i] != NULL; i++);
	b[i++] = kt;
	b[i] = NULL;
}
```

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.

3.4 Nắn tên theo quy cách

```
void NanTen(char a[MAX])
{
	int i;
	char b[MAX];//Xau trung gian
	b[0] = NULL;
	i = 0;
	//Khu dau cach
	while (a[i] == CACH)
		i++; //a[i] == NULL hay la dau 1 tu
	while (a[i] != NULL) //xu ly tu
	{
		ChenCuoi(b, Chuyen_KT_Hoa(a[i])); // xu ly dau tu: a[i]
		i++; //xet ky tu ke tiep : than tu, CACH hay ket thuc xau
		while (a[i] != CACH && a[i] != NULL) //ky tu trong than tu
		{
			ChenCuoi(b, Chuyen_KT_Thuong(a[i])); // xu ly than tu
			i++;
		} //a[i] == CACH hay a[i] == NULL
		//Da xu ly xong 1 tu
		//Chua ket thuc xau thi tiep tuc xu ly tu tiep theo
		//Tiep tuc vuot dau cach
		while (a[i] == CACH)
			i++; //a[i] == NULL hay la dau 1 tu
		if (a[i]!= NULL) //tu vua xu ly chua phai la tu cuoi
			ChenCuoi(b, CACH); // chen cac vao sau b
	}
	strcpy_s(a, MAX, b);
}
```

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.

3.5 Khai báo nguyên mẫu :

```
char Chuyen_KT_Thuong(char x);
char Chuyen_KT_Hoa(char x);
void ChenCuoi(char b[MAX], char kt);
void NanTen(char a[MAX]);
```

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.

Bước 4:

Cập nhật lại hàm ChayChuongTrinh() : Nhập chuỗi, xuất chuỗi, gọi hàm NanTen để giải quyết bài toán, điều khiển lặp công việc giải quyết bài toán.

```
void ChayChuongTrinh()
{
	char a[MAX];
	char thoat;
	do
	{
		system("CLS");
		cout << "\nNhap xau ky tu : ";
		gets_s(a);
		system("CLS");
```

```
            cout << "\nTen vua nhap: ";
            cout << a;
            NanTen(a);
            cout << "\nTen da nan  : ";
            cout << a;
            cout << endl << "Nua khong ? go ESC neu khong\n";
            thoat = _getch();
        } while (thoat != 27);
}
```

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.

Kiểm tra kết quả chương trình – Kết thúc chương trình.

E. Bài tập bắt buộc

Tất cả các bài tập phải được tổ chức dưới dạng thư viện hàm và có/không menu chức năng.

Bài 1: Xử lý chuỗi

Viết chương trình thực hiện các chức năng sau trên chuỗi:

- Thoat khoi chuong trinh
- Nhap chuoi
- Xem chuoi
- Tính chiều dài chuỗi
- Chèn ký tự x vào đầu chuỗi
- Chèn ký tự x vào cuối chuỗi
- Chen ky tu x vao chuoi a tai vi tri cho truoc
- Xóa ký tu dau chuoi
- Xóa ky tu cuoi chuoi
- Xóa ky tu tai vi tri cho truoc
- Cắt ký tự đầu chuỗi rồi chèn vào vị trí cuối chuỗi (đã cắt)
- Cắt ký tự cuối chuỗi rồi chèn vào vị trí đầu chuỗi (đã cắt)
- Xóa tất cả ký tự ***x*** cho trước khỏi chuỗi.
- Thay thể tất cả ký tự x trong chuỗi thành ký tự y.

Tạo dự án Win32 Console Application mới. Đặt tên là **Lab07_E_Bai1**

Tổ chức dự án như bài 1 mục D

(Lưu ý là nhập, xuất chuỗi không cần viết hàm, mà có thể dùng : gets_s(a,MAX), cout<<)

Tham khảo tập tin ***thuvien.h*** sau đây:

```
//Dinh nghia hang

#define MAX 100

//Dinh nghia kieu du lieu moi
typedef char String[MAX];

//Khai bao nguyen mau
int TinhChieuDaiChuoi(String a);
void ChenDau_KT(char a[MAX], char x);
```

```
void ChenCuoi_KT(char a[MAX], char x);
int ChenKT_VT(char a[MAX], char x, int vt);
void XoaDau_KT(char a[MAX]);
void XoaCuoi_KT(char a[MAX]);
int XoaKT_VT(char a[MAX], int vt);
void CatDauChenCuoi(char a[MAX]);
void CatCuoiChenDau(char a[MAX]);
void Xoa_x(char a[MAX], char x);
void Thay_x_Bang_y(char a[MAX], char x, char y);

//Dinh nghia cac ham xu ly
//Tinh chieu dai chuoi
 int TinhChieuDaiChuoi(String a)
{
	int i = 0;
	while (a[i])
		i++;
	return i;
}

 //Chen ky tu x vao dau chuoi a, ket qua luu tru vao a a
void ChenDau_KT(char a[MAX], char x)
{
	int i;
	int l = TinhChieuDaiChuoi(a);
	for (i = l; i >= 0; i--)
		a[i + 1] = a[i];
	a[0] = x;
}

//Chen ky tu x vao cuoi chuoi a, ket qua luu tru vao a a
void ChenCuoi_KT(char a[MAX], char x)
{
	int l = TinhChieuDaiChuoi(a);
	a[l++] = x;
	a[l] = NULL;
}
//Chen ky tu x vao chuoi a tai vi tri vt
//Input : a,x,vt
//output : 1; thanh cong; 0 : khong thanh cong
int ChenKT_VT(char a[MAX], char x, int vt)
{
	int i, l, kq = 1;
	l = TinhChieuDaiChuoi(a);
	if (vt == 0)
	{
		ChenDau_KT(a, x);
	}
	else
		if (vt == l)
		{
			ChenCuoi_KT(a, x);
		}
		else
```

```
                        if (0 < vt && vt < l)
                        {
                                for (i = l; i >= vt; i--)
                                        a[i + 1] = a[i];
                                a[vt] = x;
                        }
                        else
                                kq = 0;
        return kq;
}

//Xoa ky tu dau chuoi a
void XoaDau_KT(char a[MAX])
{
        int i;
        for (i = 0; a[i] != NULL; i++)
                a[i] = a[i+1];
        a[i-1] = NULL;
}

//Xoa ky tu cuoi chuoi a
void XoaCuoi_KT(char a[MAX])
{
        int i;
        for (i = 0; a[i] != NULL; i++);
        a[i - 1] = NULL;
}

//CXoa ky tu tai vi tri vt cua chuoi a
//Input : a,vt
//output : 1; thanh cong; 0 : khong thanh cong
int XoaKT_VT(char a[MAX], int vt)
{
        int i, l, kq = 1;
        l = TinhChieuDaiChuoi(a);
        if (vt == 0)
        {
                XoaDau_KT(a);
        }
        else
        if (vt == l-1)
        {
                XoaCuoi_KT(a);
        }
        else
        if (0 < vt && vt < l-1)
        {
                for (i = l-1; i > vt; i--)
                        a[i - 1] = a[i];
                a[l-1] = NULL;
        }
        else
                kq = 0;
        return kq;
```

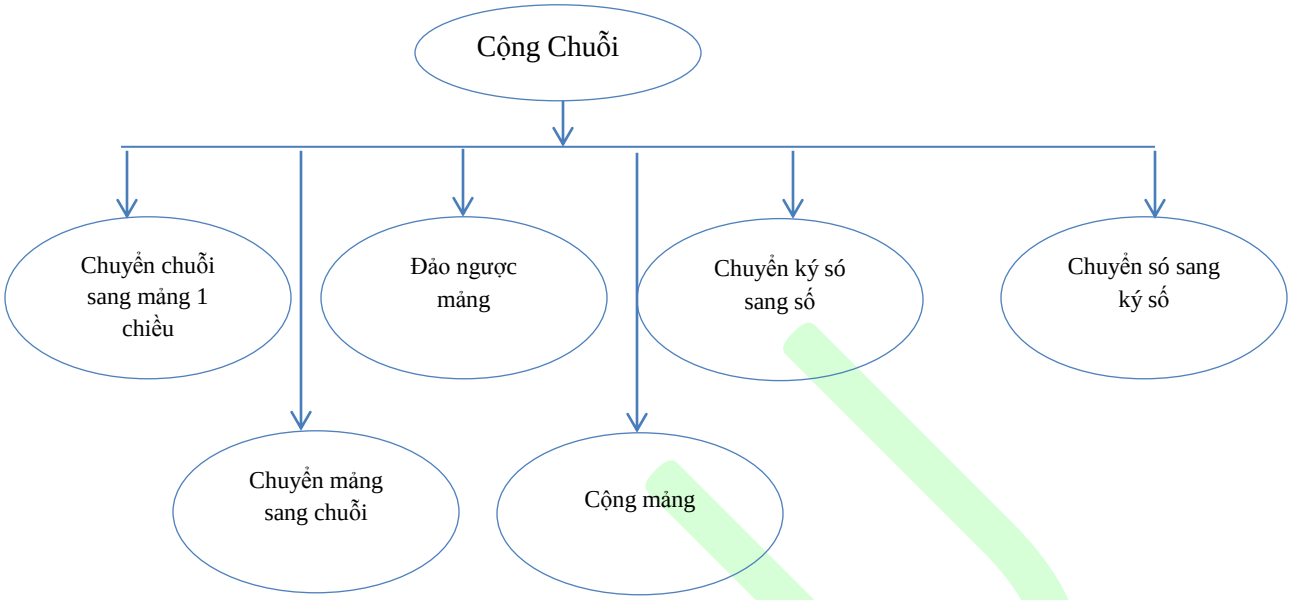
```
}

//Cat ky tu dau chuoi roi chen vao sau chuoi
void CatDauChenCuoi(char a[MAX])
{
	int i;
	char x;
	x = a[0];
	for (i = 1; a[i] != NULL; i++)
		a[i - 1] = a[i];
	a[i - 1] = x;
}

//Cat ky tu cuoi chuoi roi chen vao tai vi tri dau chuoi
void CatCuoiChenDau(char a[MAX])
{
	int i, l;
	char x;
	l = TinhChieuDaiChuoi(a);
	x = a[l-1];
	for (i = l-2; i >= 0; i--)
		a[i+1] = a[i];
	a[0] = x;
}
//Xoa tat ca ky tu x trong chuoi
void Xoa_x(char a[MAX], char x)
{
	int i, h=0;
	for (i = 0; a[i] != NULL; i++)
		if (a[i] != x)
			a[h++] = a[i];
		a[h] = NULL;
}
//Thay the tat ca cac ky tu x trong chuoi bang ky tu y
void Thay_x_Bang_y(char a[MAX], char x, char y)
{
	int i;
	for (i = 0; a[i] != NULL; i++)
	if (a[i] == x)
		a[i] = y;
}
```

Bài 2. Phép cộng số tự nhiên lớn

Cho hai số tự nhiên lớn ***m***, ***n*** (không quá 50 chữ số) . Viết chương trình thực hiện phép cộng hai số ***m***, ***n***.

Ý tưởng giải quyết vấn đề :

- ***Số tự nhiên lớn sẽ được biểu diễn bởi xâu ký tự. Và ta phải thực hiện phép cộng các xâu ký số.***
- ***Để đơn giản, thay vì ta thực hiện phép công các xâu ký số ta chuyển sang thực hiện phép cộng 2 mảng 1 chiều. Nên ta phải chuyển cấu trúc xâu sang mảng 1 chiều.***
- ***Khi cộng mảng, nếu xuất phát từ cách thức cộng thông thường thì ta thực hiện từ phải sang trái, khi đó sẽ vấp phải trường hợp tràn ô nhớ, tức là ở hang cao nhất mà có nhớ thì không có nơi để lưu trữ. Để khắc phục điều này, ta đảo ngược mảng và thực hiện từ trái sang phải.***



Tổ chức chương trình bài 2 phần D (hướng dẫn), không có hệ thống menu.

Tham khảo tập tin ***thuvien.h*** sau đây :

```
//Dinh nghia hang
#define MAX 50
//Dinh nghia kieu du lieu moi
typedef char SoTuNhienLon[MAX];
//============================
int Chuyen_KySo_So(char x)
{
	return x - '0';
}
char Chuyen_So_KySo(int so)
{
	char x = (char)(so + '0');
	return x;
}

//Chuyen chuoi sang mang 1 chieu
void Chuyen_Chuoi_Sang_Mang(SoTuNhienLon m, int a[MAX], int &l)
{
	int i;
	l = strlen(m);
	for (i = 0; i < l; i++)
		a[i] = Chuyen_KySo_So(m[i]);
}

//Dao nguoc mang
void DaoNguocMang(int a[MAX], int l)
{
	int i, j, tam;
	for (i = 0, j = l - 1; i < j; i++, j--)
	{
		tam = a[i];
```

```
                a[i] = a[j];
                a[j] = tam;
        }
}

//Chuyen mang sang chuoi
void ChuyenMang_Sang_Chuoi(int a[MAX], int l, SoTuNhienLon m)
{
        int i;
        for (i = 0; i < l; i++)
                m[i] = Chuyen_So_KySo(a[i]);
        m[l] = NULL;
}

//Cong 2 mang
void CongMang(int a[MAX], int la, int b[MAX], int lb, int c[MAX], int &lc)
{
        int nho , i;
        lc = la >= lb ? la : lb;

        if (lc > lb)
                for (i = lb; i < lc; i++)
                        b[i] = 0;
        if (lc > la)
                for (i = la ; i < lc; i++)
                        a[i] = 0;

        nho = 0;
        for (i = 0; i < lc; i++)
        {
                c[i] = a[i] + b[i] + nho;
                if (c[i] > 9)
                {
                        c[i] = c[i] - 10;
                        nho = 1;
                }
                else
                        nho = 0;  // tra lai nho = 0;
        }
        if (nho == 1)
        {
                c[lc] = nho;
                lc++;
        }
}
//Cong 2 so tu nhien lon
void CongSoTuNhienLon(SoTuNhienLon m, SoTuNhienLon n, SoTuNhienLon t)
{
        int a[MAX], b[MAX], c[MAX];
        int la, lb, lc;
        Chuyen_Chuoi_Sang_Mang(m, a, la);
        Chuyen_Chuoi_Sang_Mang(n, b, lb);

        DaoNguocMang(a, la);
```

```
        DaoNguocMang(b, lb);

        CongMang(a, la, b, lb, c, lc);

        DaoNguocMang(c, lc);
        ChuyenMang_Sang_Chuoi(c, lc, t);
}
```

Bài 3. Xử lý chuỗi

Viết chương trình thực hiện các chức năng sau trên chuỗi :

- Chuyển tất cả các ký tự trong chuỗi thành ký tự thường
- Chuyển tất cả các ký tự trong chuỗi thành ký tự HOA
- Đảo ngược chuỗi
- Kiểm tra một chuỗi có phải là chuỗi đối xứng (Palindrome).

Bài 4. Tìm – Đếm

Viết chương trình thực hiện các thao tác sau trên chuỗi :

- Đếm số lượng ký tự x xuất hiện trong chuỗi
- Xuất các giá trị ký tự phân biệt của chuỗi và số lần xuất hiện tương ứng của nó.
- Tìm vị trí xuất hiện đầu tiên của ký tự ***x*** trong chuỗi. Nếu không trả về -1
- Tìm vị trí xuất hiện của chuỗi ***t*** trong chuỗi ***s***. Nếu ***s*** không chứa ***t*** thì trả về -1.
- Đếm số từ trong chuỗi ***s***.
- Đảo vị trí của từ đầu và từ cuối trong chuỗi ***s***. Ví dụ: s = meo an ca thì kết quả là s = ca an meo

Bài 5. Chuyển xâu ký số thành số tự nhiên

Viết chương trình đổi một xâu ký số thành giá trị số tương ứng.

Ví dụ : nhập "123456567890", Xuất : 1234567890

F. Bài tập làm thêm

Bài 1. Từ chung

Viết chương trình thực hiện các chức năng sau trên chuỗi ký tự:

- Liệt kê các ký tự xuất hiện trong cả hai chuỗi ***s*** và ***t***.
- Liệt kê các từ xuất hiện trong cả hai chuỗi ***s*** và ***t***.
- Tìm và liệt kê từ dài nhất trong chuỗi ***s***.

Bài 2. Tách từ

Viết chương trình thực hiện các chức năng sau trên chuỗi ký tự:

- Tách chuỗi s thành một mảng các từ dựa vào khoảng trắng. Ví dụ: s = "Khoa Cong nghe Thong tin" thì tách thành mảng gồm các chuỗi: "Khoa", "Cong", "nghe", "Thong", "Tin".
- Kiểm tra chuỗi s có chứa ký tự số hay không? Nếu có, tách các số đó ra thành một mảng số riêng. Ví dụ: s = "Thang 1 Nam 2015, dan so Viet Nam vao khoang 90 trieu" thì mảng số kết quả gồm các phần tử: 1, 2015, 90.
- Đảo ngược thứ tự các từ trong chuỗi ***s***. Ví dụ: "Thuong qua Viet Nam" => "Nam Viet qua Thuong"

Bài 3. Năm Dương lịch – Âm Lịch

Viết chương trình nhập vào năm Dương lịch, xuất ra năm Âm lịch

Ví dụ:

Input : 2016

Output: Binh Than

Bài 4. Nén xâu

Viết chương trình thực hiện việc nén và giải nén một xâu ký tự. Ví dụ: xâu s = "AAABBCCCCC CDDDEEEEEEEFFFG" sau khi nén là s = "3A2B6C3D7E3FG". *Chú ý: nếu ký tự chỉ xuất hiện 1 lần thì không cần nén, không cần ghi số lần xuất hiện ký tự đó. Trong trường hợp này là ký tự G.*

Cho biết cách nén xâu theo cách trên không có tác dụng trong những trường hợp nào?

Bài 5. Mã hóa

Viết chương trình mã hóa và giải mã một chuỗi ký tự bằng cách đảo bit (0->1, 1->0) hoặc đảo ngược thứ tự các bit của từng ký tự trong chuỗi.