

LAB 4. TỔ CHỨC THƯ VIỆN HÀM VÀ MENU

THỜI LƯỢNG: 6 TIẾT

A. Mục tiêu

- Giúp sinh viên làm quen với việc tổ chức chương trình theo các thư viện hàm (*.h).
- Hướng dẫn sinh viên cách tạo menu để người dùng chọn chức năng của chương trình.
- Tiếp tục rèn luyện kỹ năng cài đặt chương trình theo cách từng bước bổ sung các chức năng vào chương trình và viết chương trình có cấu trúc hình thức.
- Sau khi hoàn thành bài thực hành này, sinh viên cần phải:
 - Tạo được dự án dạng Win32 Console Application có sử dụng các tập tin header (*.h).
 - Sử dụng thuần thục các cấu trúc điều khiển, định nghĩa kiểu dữ liệu, biến, hằng,...
 - Sử dụng được hàm.
 - Hiểu rõ truyền tham số bằng trị
 - Biết cách tổ chức chương trình theo thư viện hàm, hệ thống menu.

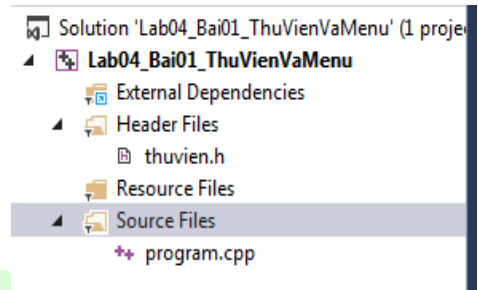
B. Yêu cầu

- Trong phần phần C (hướng dẫn thực hành), sinh viên tự đọc mục 1, mục 2 (hướng dẫn tổ chức chương trình)..
- Nộp kết quả thực tập phần C (hướng dẫn thực hành) tại phòng Lab theo yêu cầu :
 - Tạo thư mục, đặt tên là **MSSV_Lab04_C_HD**, để lưu bài làm. Trong đó, MSSV là mã số của sinh viên.
 - Các bài 1,2 3 : tạo các project theo hướng dẫn, lưu trữ trong thư mục trên
 - Xóa thư mục Debug trong các project, nén thư mục và nộp cho giáo viên qua mail.
 - Thời gian nộp : Cuối tiết 2 buổi thực tập thứ 5
- Sinh viên tiếp tục nộp kết quả thực tập (phần D, bài tập bắt buộc) tại phòng Lab theo yêu cầu :
 - Tạo thư mục, đặt tên là **MSSV_Lab04_D_BB**, để lưu bài làm. Trong đó, MSSV là mã số của sinh viên.
 - Thực hiện các bài tập 1,2 trong phần D, mỗi bài tập là một project.
 - Xóa thư mục Debug trong các project, nén thư mục và nộp cho giáo viên qua mail.
 - Thời gian nộp : Cuối tiết 4 buổi thực tập thứ 5.
 -
- Sinh viên tiếp tục nộp kết quả thực tập (phần D, bài tập bắt buộc) tại phòng Lab theo yêu cầu :
 - Tạo thư mục, đặt tên là **MSSV_Lab04_D_BB_TT**, để lưu bài làm. Trong đó, MSSV là mã số của sinh viên.
 - Thực hiện các bài tập 3,4,5 trong phần D, mỗi bài tập là một project.
 - Xóa thư mục Debug trong các project, nén thư mục và nộp cho giáo viên qua mail.
 - Thời gian nộp : Cuối tiết 2 buổi thực tập thứ 6

C. Hướng dẫn thực hành

1. Tạo cấu trúc cho chương trình có sử dụng thư viện hàm do người dùng định nghĩa

Phần này hướng dẫn cách tạo một dự án có sử dụng tập tin tiêu đề hay tập tin thư viện hàm (*.h). Các tập tin này được chứa trong thư mục Header Files của dự án. Hình bên cho thấy cấu trúc của dự án mà ta sẽ tạo.



- Tập tin **program.cpp** chứa hàm main, và hàm ChayChuongTrinh(). Hàm main() là nơi khởi đầu của chương trình, sẽ gọi hàm ChayChuongTrinh() để thực hiện chương trình, còn hàm ChayChuongTrinh() thực hiện các chức năng của chương trình.
- Tập tin **thuvien.h** sẽ chứa phần định nghĩa kiểu dữ liệu mới, hằng số và các hàm xử lý (tùy thuộc bài toán).

Từ đây trở đi, mỗi khi xây dựng dự án mới (có tổ chức thư viện, không có hệ thống menu tùy chọn), cần tạo cấu trúc cho chương trình như hình trên.

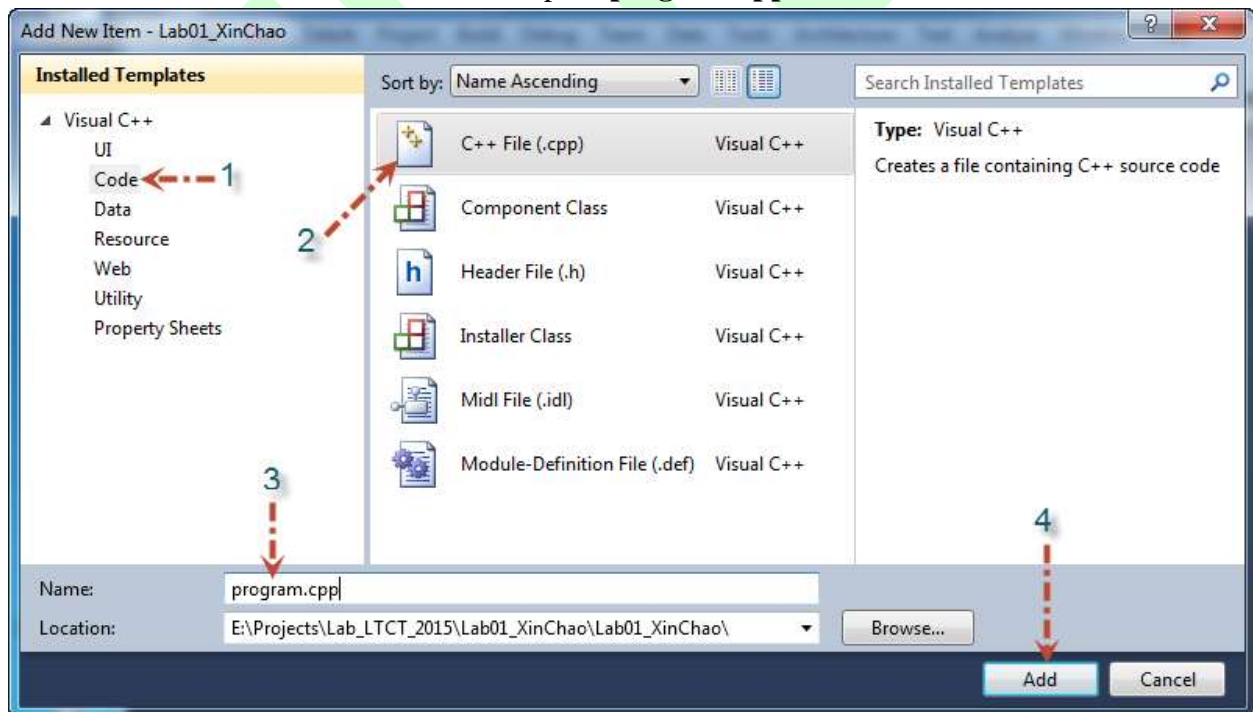
Theo cách tổ chức này, tập tin program.cpp và hàm main rất ít khi thay đổi mã lệnh.

Bước 1. Tạo dự án Win32 Console Application mới. Đặt tên là **Lab04_C_Muc1_ThuVien**.

Bước 2. Sử dụng 1 trong 3 cách sau để mở cửa sổ **Add New Item**

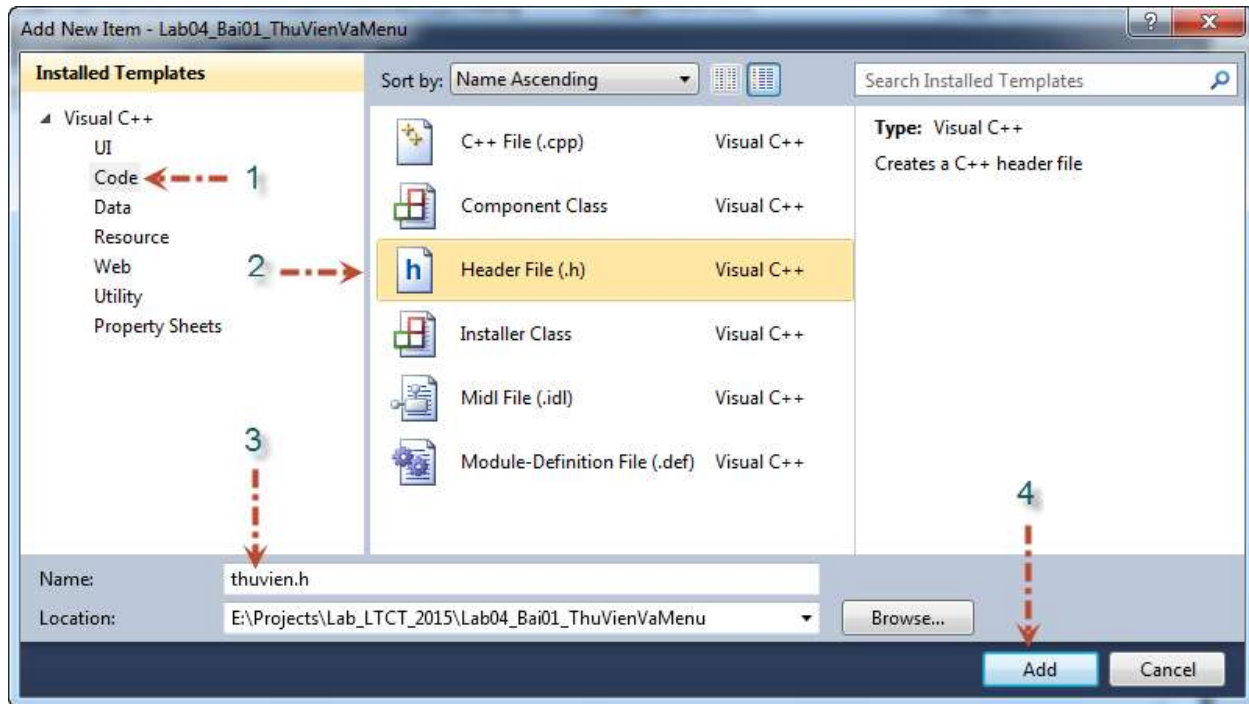
- Chọn thư mục **Source Files**, nhấn tổ hợp phím Ctrl + Shift + A
- Nhấp phải thư mục **Source Files** trong dự án, chọn Add > New Item ...
- Chọn thư mục **Source Files**, sau đó chọn menu Project > Add New Item...

Bước 3. Trong cửa sổ **Add New Item**, chọn mục **Code > C++ File (.cpp)**, đặt tên là **program.cpp** trong mục Name. Nhấn nút Add để thêm tập tin **program.cpp** vào dự án.



Bước 4. Thực hiện như bước 2 để mở cửa sổ Add New Item nhưng chọn thư mục **Header Files**.

Bước 5. Trong cửa sổ **Add New Item**, chọn mục **Code > Header File (.h)**, đặt tên là **thuvien.h** trong mục Name. Nhấn nút Add để thêm tập tin **thuvien.h** vào dự án.



Các bước tiếp theo sẽ soạn thảo nội dung cho các tập tin *Program.cpp*, *thuvien.h*; Lưu ý là khi soạn thảo nội dung chương trình, về mặt hình thức văn bản phải có cấu trúc : các câu lệnh không phụ thuộc nhau sẽ viết trên cùng một cột, nếu câu lệnh sau phụ thuộc câu lệnh trước thì dịch câu lệnh sau vào một tab,...Khi soạn nội dung cho tập tin, câu lệnh đầu tiên sẽ bắt đầu từ hàng đầu, cột đầu.

Bước 6. Trong tập tin *Program.cpp*, nhập đoạn code sau (phần tối thiểu để có thể chạy được chương trình)

```
//Chen cac tap tin thu vien co san
#include <iostream>
#include <conio.h>
using namespace std;
//Chen cac tap tin thu vien tu tao : luu y tring tu chen.
#include "Thuvien.h"
//khai bao nguyen mau ham
void ChayChuongTrinh();
int main()
{
    ChayChuongTrinh();
    return 1;
}
void ChayChuongTrinh()
{
    _getch();
}
```

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

Sau đây, ta từng bước bổ sung chức năng để hoàn thiện chương trình, mỗi lần bổ sung xong ta chạy kiểm tra chương trình để sửa lỗi.

Bước 7. Trong tập tin *thuvien.h*, nhập nội dung như sau:

```

1 // =====
2 // Định nghĩa các hằng số và kiểu dữ liệu mới
3 // =====
4
5
6 // =====
7 // Khai báo nguyên mẫu các hàm xử lý
8 // =====
9
10
11 // =====
12 // Định nghĩa các hàm xử lý
13 // =====
14
15 ~

```

Nội dung tập tin `thu_vien.h` sẽ bao gồm các phần : định nghĩa hằng, kiểu dữ liệu mới, định nghĩa các hàm chức năng của chương trình.

Mỗi lần bổ sung đoạn lệnh (định nghĩa hằng, kiểu dữ liệu mới, hàm xử lý,...) ta đều phải kiểm tra chương trình để sửa lỗi nếu có.

Bước 8. Trong tập tin `program.cpp`, ta sửa lại hàm `ChayChươngTrinh()` để thực hiện các công việc của chương trình.

Bước 9. Biên dịch và chạy chương trình để xem kết quả.

Hãy lưu ý việc viết chương trình theo cách hoàn thiện dần từng bước (tại mỗi bước đều có chạy kiểm tra chương trình để sửa lỗi nếu có).

Bài 1 : Trò chơi đoán số

Viết chương trình thực hiện trò chơi đoán số dưới dạng tổ chức thư viện.

Gợi ý tổ chức chương trình như sau :

- Viết hàm cho người dùng chọn mức số lần đoán.
- Viết hàm sinh số ngẫu nhiên (Đề : Số do máy tính nghĩ ra làm đề)
- Viết hàm xử lý trò chơi : Kết thúc trò chơi hàm trả về 1 (người chơi thắng) hoặc 0 (người chơi thua). Tại mỗi lần đoán sai, chương trình có thông báo hướng dẫn theo yêu cầu bài toán.
- Viết hàm thông báo kết quả thắng thua (lưu ý rằng cần phải xuất số đề ra màn hình).
- Viết hàm `ChayChươngTrinh()` điều khiển thực hiện trò chơi.
- Có thể điều khiển chương trình cho phép chơi nhiều lần cho đến khi người dùng không muốn chơi nữa thì dừng.

Bước 1. Tạo dự án Win32 Console Application mới. Đặt tên là **Lab04_C_Bai01_TroChoiDoanSo**.

Bước 2. Tạo cấu trúc chương trình như đã hướng dẫn trong **mục 1** (từ bước 1 đến bước 7).

Bước 3. Trong tập tin `thuvien.h`, từng bước bổ sung cài đặt các hàm xử lý :

//Định nghĩa hằng

```
#define MAX 1000 //giới hạn số dương ngẫu nhiên máy tính nghĩ ra < MAX.
```

```
//Định nghĩa kiểu dữ liệu mới – không có
```

```
//Khai báo nguyên mẫu các hàm xử lý
```

```
//... (bổ sung sau)
```

//Định nghĩa các hàm xử lý

3.1 Hàm chọn số lần đoán (Một lần chơi được phép đoán bao nhiêu lần)

```
//Input : không có
```

```
//Output : k = số lần đoán ( giả sử có 3 mức tương ứng với số lần đoán là : 3,5,7)
```

```
int ChonMucDoan()
{
    int k;
    do
    {
        cout << "\nChon so lan doan toi da ( 3,5,7) : k = ";
        cin >> k;
    } while (k != 3 && k != 5 && k != 7);
    return k;
}
```

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

3.2 Hàm sinh số ngẫu nhiên

```
//Input : không có
```

```
//Output : soDe = một số dương ngẫu nhiên < MAX
```

```
int SinhSoNgauNhiem()
{
    int soDe;
    //gieo so ngau nhien
    srand((unsigned int)time(0)); //khai bao <time.h>
    //tao so ngau nhien
    soDe = rand()/MAX; //khai bao <stdlib.h>
    return soDe;
}
```

- Trong tập tin **Program.cpp**, khai báo bổ sung các thư viện :

```
#include <time.h>
#include <stdlib.h>
```

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

3.3. Viết hàm xử lý trò chơi

```
//Input: k = số lần đoán
//      soDe = đề do máy tính nghĩ ra
//Output : 1 (người chơi thắng)
//        0 (người chơi thua)
```

```
int XuLyTroChoi(int k, int soDe)
{
    int i,
        soDoan,
        kq = 0;
    for (i = 1; i <= k; i++)
    {
        cout << "\nDoan lan " << i << ", so doan = ";
        cin >> soDoan;
        if (soDoan == soDe)
        {
            kq = 1;
            break;
        }
        else
        if (soDoan > soDe)
            cout << "\nSo doan lon hon";
        else
            cout << "\nSo doan nho hon";
    }
    return kq;
}
```

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

3.4 Viết hàm thông báo kết quả trò chơi

```
//Input : kq,
//      soDe;
//Output : không có
```

```
void ThongBaoKetQua(int kq, int soDe)
{
    system("CLS");
    cout << "\nKET QUA TRO CHOI : ";
    if (kq)
        cout << "\nNguoi chơi thắng";
    else
        cout << "\nNguoi chơi thua";
    cout << "\nĐề cho số : " << soDe;
}
```

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

3.5. Bổ sung khai báo nguyên mẫu các hàm (tại vùng khai báo nguyên mẫu hàm):

```
int ChonMucDoan();
```

```
int SinhSoNgauNhiem();
int XuLyTroChoi(int k, int soDe);
void ThongBaoKetQua(int kq, int soDe);
```

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

Bước 6: Trong tập tin **Program.cpp**, cập nhật lại hàm ChayChuongTrinh() như sau để hoàn thiện chương trình.

- Hàm ChayChuongTrinh() sẽ điều khiển lặp lại việc chơi trò chơi.

```
void ChayChuongTrinh()
{
    char kt;
    int kq,k, soDe;
    do
    {
        system("CLS");
        soDe = SinhSoNgauNhiem();
        k = ChonMucDoan();
        kq = XuLyTroChoi(k, soDe);
        system("CLS");
        cout << "\nTRO CHOI DOAN SO VOI SO LAN DOAN : k = " << k << " : \n";
        ThongBaoKetQua(kq, soDe);
        _getch();
        system("CLS");
        cout << "\nChoi nua khong, nhan ESC neu khong!\n";
        kt = _getch();
    } while (kt != 27);
}
```

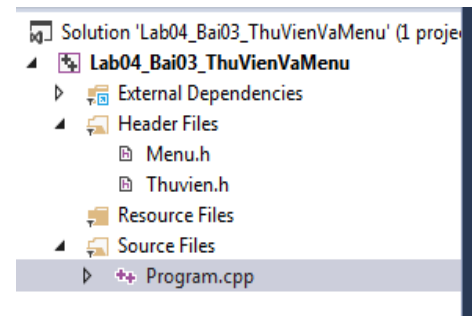
- Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.

- Thực hiện trò chơi, kiểm tra kết quả có phù hợp hay không ?

2. Tạo cấu trúc cho chương trình có sử dụng thư viện hàm do người dùng định nghĩa, và có tổ chức hệ thống menu :

Phần này hướng dẫn cách tạo một dự án tổ chức theo thư viện hàm (*.h) và hệ thống menu tùy chọn. Các tập tin này được chứa trong thư mục Header Files của dự án. Hình bên cho thấy cấu trúc của dự án mà ta sẽ tạo.

- Tập tin **program.cpp** chứa hàm main, và hàm ChayChuongTrinh(). Hàm main() là nơi khởi đầu của chương trình, sẽ gọi hàm ChayChuongTrinh() để thực hiện chương trình, còn hàm ChayChuongTrinh() điều khiển và thực hiện các chức năng của chương trình.
- Tập tin **thuvien.h** sẽ chứa phần định nghĩa kiểu dữ liệu mới, hằng số và các hàm xử lý (tùy thuộc bài toán).



- Tập tin **menu.h** chứa các hàm để xử lý menu chức năng do người dùng chọn.

Từ đây trở đi, mỗi khi xây dựng dự án mới tổ chức theo thư viện và có hệ thống tùy chọn menu, sẽ tạo cấu trúc cho chương trình như trên.

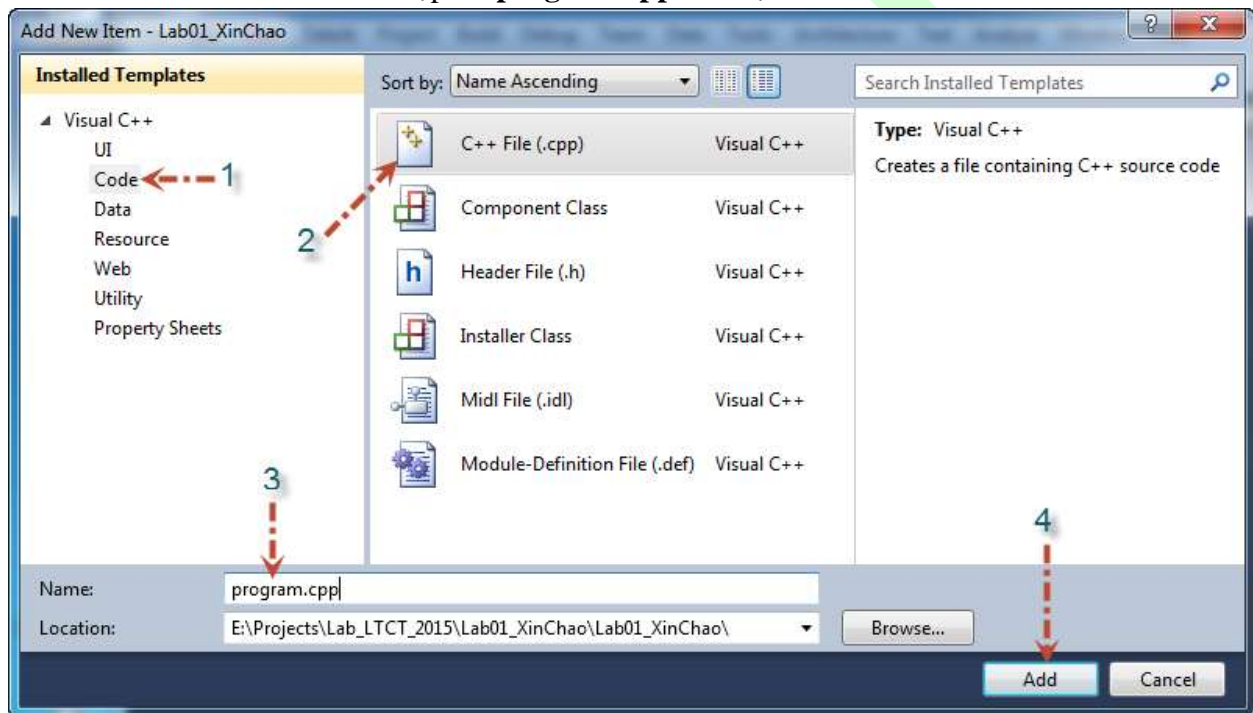
Theo cách tổ chức này, tập tin `program.cpp` và hàm `main` rất ít khi thay đổi mã lệnh.

Bước 1: Tạo dự án Win32 Console Application mới. Đặt tên là **Lab04_C_Muc2_ThuVienVaMenu**.

Bước 2: Sử dụng 1 trong 3 cách sau để mở cửa sổ **Add New Item**

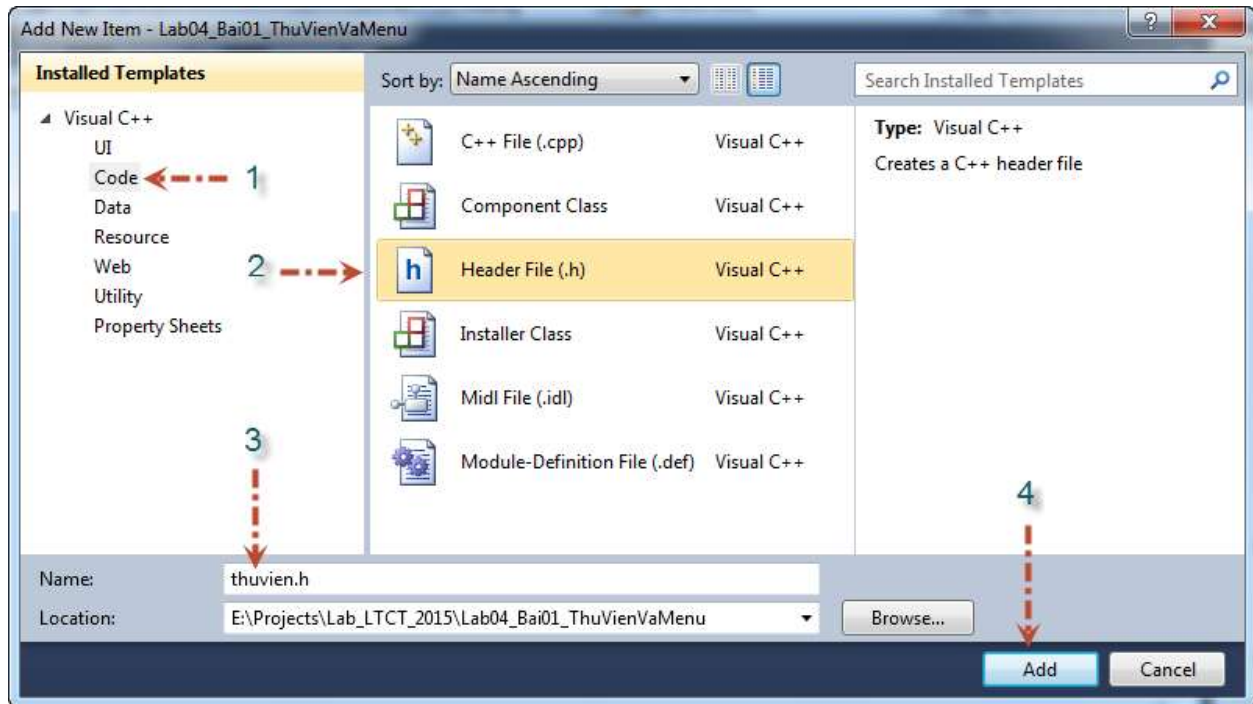
- Chọn thư mục **Source Files**, nhấn tổ hợp phím Ctrl + Shift + A
- Nhấp phải thư mục **Source Files** trong dự án, chọn Add > New Item ...
- Chọn thư mục **Source Files**, sau đó chọn menu Project > Add New Item...

Bước 3: Trong cửa sổ **Add New Item**, chọn mục **Code > C++ File (.cpp)**, đặt tên là **program.cpp** trong mục Name. Nhấn nút Add để thêm tập tin **program.cpp** vào dự án.



Bước 4: Thực hiện như bước 2 để mở cửa sổ Add New Item nhưng chọn thư mục **Header Files**.

Bước 5: Trong cửa sổ **Add New Item**, chọn mục **Code > Header File (.h)**, đặt tên là **thuvien.h** trong mục Name. Nhấn nút Add để thêm tập tin **thuvien.h** vào dự án.



Bước 6: Làm lại bước 4 và 5 ở trên nhưng đặt tên tập tin là **menu.h**.

Bước 7: Trong tập tin **Program.cpp**, nhập đoạn code sau (phần tối thiểu để có thể chạy được chương trình :

//Chen cac tap tin thu vien co san

#include <iostream>

#include <conio.h>

using namespace std;

//Chen cac tap tin thu vien tu tao : luu y trinh tu chen.

#include "Thuvien.h"

#include "Menu.h"

//khai bao nguyen mau ham

void ChayChuongTrinh();

int main()

{

 ChayChuongTrinh();

 return 1;

}

void ChayChuongTrinh()

{

 _getch();

}

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

Sau đây, ta bổ sung từng bước chức năng để hoàn thiện chương trình, mỗi lần bổ sung ta phải chạy kiểm tra chương trình để sửa lỗi.

Bước 8: Trong tập tin **thuvien.h**, nhập nội dung như sau:

```

1 // =====
2 // Định nghĩa các hằng số và kiểu dữ liệu mới
3 // =====
4
5
6 // =====
7 // Khai báo nguyên mẫu các hàm xử lý
8 // =====
9
10
11 // =====
12 // Định nghĩa các hàm xử lý
13 // =====
14
15 ~

```

Bước 9: Trong tập tin **menu.h**, nhập đoạn mã sau:

```

1 //=====
2 //Khai báo nguyên mẫu các hàm trong hệ thống menu
3 //=====
4 void XuatMenu();
5 int ChonMenu(int soMenu);
6 void XuLyMenu(int menu);
7
8 //=====
9 //Định nghĩa các hàm trong hệ thống menu
10 //=====
11 //Hàm XuatMenu : Xuất danh sách các chức năng ra màn hình:
12 // Gia Su có 4 chức năng
13 //Input : Không có
14 //Output : Không có
15 void XuatMenu()
16 {
17     cout << "\n===== HE THONG CHUC NANG =====";
18     cout << "\n0. THOAT KHOI CHUONG TRINH";
19     cout << "\n1. chuc nang 1";
20     cout << "\n2. chuc nang 2";
21     cout << "\n3. chuc nang 3";
22     cout << "\n4. chuc nang 4";
23     cout << "\n===== ";
24 }
25
26 //Hàm ChonMenu : Chọn một chức năng trong danh sách
27 //Input : soMenu = Số chức năng
28 //Output : stt == thứ tự chức năng được chọn

```

```

29 int ChonMenu(int soMenu)
30 {
31     int stt; //bien luu so thu tu chuc nang duoc chon
32     for (;;)
33     {
34         system("CLS");
35         XuatMenu();
36         cout << "\nNhap 1 so stt (1 <= stt <= " << soMenu
37             << ") de chon menu : stt = ";
38         cin >> stt;
39         if (0 <= stt && stt <= soMenu)
40             break;
41     }
42     return stt;
43 }
44
45 //Ham XuLyMenu : Xu ly chuc nang duoc chon
46 //Input : menu = So thu tu menu da chon
47 //Ouput : khong co
48 void XuLyMenu(int menu)
49 {
50     switch (menu)
51     {
52     case 0:
53         cout << "\n0. THOAT KHOI CHUONG TRINH.\n";
54         break;
55     case 1:
56         cout << "\n1. Ban da chon chuc nang 1";
57         //Xu ly chuc nang 1
58         break;
59     case 2:
60         cout << "\n2. Ban da chon chuc nang 2";
61         //Xu ly chuc nang 2
62         break;
63     case 3:
64         cout << "\n3. Ban da chon chuc nang 3";
65         //Xu ly chuc nang 3
66         break;
67     case 4:
68         cout << "\n4. Ban da chon chuc nang 4";
69         //Xu ly chuc nang 4
70         break;
71     }
72     _getch();
73 }

```

Bước 10. Trong tập tin **program.cpp**, ta sửa lại hàm **ChayChuongTrinh()** :

```

1 //=====
2 //Nạp các tập tin thư viện hàm xây dựng sẵn
3 //=====
4 #include <iostream>
5 #include <conio.h>
6 using namespace std;
7 #include "Thuvien.h"
8 #include "Menu.h"
9 //Xây dựng thêm hàm Chạy chương trình
10 void ChayChuongTrinh();
11 int main()
12 {
13     ChayChuongTrinh();
14     return 1;
15 }
16 void ChayChuongTrinh()
17 {
18     int menu,
19     soMenu = 4;
20     do
21     {
22         menu = ChonMenu(soMenu);
23         XuLyMenu(menu);
24     } while (menu > 0);
25 }

```

Bước 11. Biên dịch và chạy chương trình để xem kết quả.

Lưu ý quan trọng

- Tùy thuộc vào chương trình, ta cần thay đổi các hàm main, XuatMenu, XuLyMenu.
- Để đổi tên chức năng hoặc thêm các chức năng, viết lại các lệnh cout trong hàm **XuatMenu**.
- Để thay đổi cách xử lý từng menu, viết lại mã lệnh trong hàm **XuLyMenu**.
- Khi số menu thay đổi, gán giá trị mới cho biến soMenu trong hàm **ChayChuongTrinh**.
- Nếu tất cả các chức năng đều sử dụng chung một đầu vào thì có thể bổ sung các tham số tương ứng cho hàm **XuLyMenu**.

Bài 2: Tính diện tích hình học

Bước 1: Tạo dự án Win32 Console Application mới. Đặt tên là **Lab04_C_Bai2_DienTichHinhHoc**.

Bước 2: Tạo cấu trúc chương trình như đã hướng dẫn trong **mục 2** (từ bước 1 đến bước 8).

Bước 3: Trong tập tin menu.h ta viết lại như sau (cấu trúc giống như bước 9 mục 2, chỉ thay đổi nội dung theo yêu cầu bài toán) :

```

// =====
// 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

// Input : Không có

// Output: Không có

```
void XuatMenu()
{
    cout << endl << "===== CHON CHUC NANG =====";
    cout << endl << "0. Thoat khoi chuong trinh";
    cout << endl << "1. Tinh dien tich hinh vuong";
    cout << endl << "2. Tinh dien tich hinh chu nhat";
    cout << endl << "3. Tinh dien tich hinh tam giac";
    cout << endl << "4. Tinh dien tich hinh tron";
    cout << endl << "===== ";
}
```

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

3.2 Định nghĩa hàm chọn một menu trong danh sách:

Hàm này điều khiển người dùng trong việc chọn chức năng. Người dùng chỉ được phép nhập số thứ tự menu trong khoảng [0,...,soMenu].

// 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");
        XuaMenu();
        cout << "\nNhap 1 so de chon menu ( 0 <= stt <= " << soMenu << " ) : 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 tùy thuộc vào chức năng:

// Input : menu = Số thứ tự menu do người chọn.

// Output: Không có.

```
void XuLyMenu(int menu)
{
    switch (menu)
    {
        case 0:
```

```

        cout << endl << "0. Thoat khoi chuong trinh";
        break;
    case 1:
        cout << endl << "1. Tinh dien tich hinh vuong";
        //bổ sung sau
        break;
    case 2:
        cout << endl << "2. Tinh dien tich hinh chu nhac";
        //bổ sung sau
        break;
    case 3:
        cout << endl << "3. Tinh dien tich hinh tam giac";
        //bổ sung sau
        break;
    case 4:
        cout << endl << "4. Tinh dien tich hinh tron";
        //bổ sung sau
        break;
    }
}

```

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);

```

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

Trong bước tiếp theo, ta soạn thảo từng phần (chức năng) trong *thuvien.h*, bổ sung xử lý các chức năng trong hàm *XuLyMenu* của *menu.h*, viết lại hàm *ChayChuongTrinh* trong *program.cpp* để điều khiển tùy chọn thực hiện menu chương trình (chọn 0 là dừng chương trình)

Bước 4 :

4.1 Trong tập tin *program.cpp*, ta cập nhật lại hàm *ChayChuongTrinh* :

```

// Định nghĩa hàm lặp lại việc chọn menu, xử lý menu
// cho tới khi người dùng chọn menu 0 thì thoát CT.
void ChayChuongTrinh()
{
    // Khai báo biến
    int menu,           // lưu số thứ tự menu được chọn
        soMenu = 4;    // lưu số lượng menu chức năng.

    // Lặp lại việc chọn và xử lý menu cho tới khi
    // người dùng chọn chức năng 0. Thoát khỏi CT.
    do
    {

```



```

        menu = ChonMenu(soMenu);
        XuLyMenu(menu);
    } while (menu > 0);
}

```

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

Kiểm tra thực hiện chức năng: chọn 0 thì dừng chương trình, chọn từ 1 đến 4 thì không làm gì cả.

Bước 5:

5.1 Trong **thuvien.h**, bổ sung:

- Trong phần định nghĩa nghĩa các hàm xử lý, ta định nghĩa hàm tính diện tích hình vuông.

// Định nghĩa hàm tính diện tích hình vuông

// Input : canh = Độ dài cạnh hình vuông.

// Output: Diện tích hình vuông

```

double DienTichHinhVuong(double canh)
{
    return canh * canh;
}

```

- Trong phần khai báo nguyên mẫu hàm, bổ sung khai báo nguyên mẫu hàm tính diện tích hình vuông :

double DienTichHinhVuong(**double** canh);

5.2 Trong tập tin **menu.h** : ta bổ sung xử lý thực hiện tính diện tích hình vuông trong case 1 của hàm XuLyMenu. Trong hàm này ta khai báo thêm các biến thực a (để lưu trữ giá trị cạnh), dienTich để lưu trữ giá trị diện tích hình vuông :

```

void XuLyMenu(int menu)
{
    // Khai báo các biến
    double a, dienTich;
    switch (menu)
    {
        case 0:
            cout << endl << "0. Thoat khoi chuong trinh";
            break;

        case 1:
            cout << endl << "1. Tinh dien tich hinh vuong";
            // Thông báo người dùng nhập độ dài cạnh
            cout << endl << "Nhap do dai canh hinh vuong : ";
            cin >> a;

            // Gọi hàm tính diện tích hình vuông
            dienTich = DienTichHinhVuong(a);

            // Xuất kết quả
            cout << endl << "Dien tich hinh vuong canh "
                << a << " la " << dienTich;
            break;

        case 2:
            cout << endl << "2. Tinh dien tich hinh chu nhac";

```

```

        //bổ sung sau
        break;
    case 3:
        cout << endl << "3. Tính diện tích hình tam giác";
        //bổ sung sau
        break;
    case 4:
        cout << endl << "4. Tính diện tích hình tròn";
        //bổ sung sau
        break;
    }
}

```

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

Tiếp theo bổ sung các chức năng khác.

Bước 6 :

6..1 Trong *thuvien.h*, bổ sung :

- Trong phần định nghĩa nghĩa các hàm xử lý, ta định nghĩa hàm tính diện tích hình chữ nhật :

```

// Định nghĩa hàm tính diện tích hình chữ nhật
// Input : dai = Kích thước chiều dài của HCN.
//         rong = Kích thước chiều rộng của HCN
// Output: Diện tích hình chữ nhật
double DienTichHinhChuNhat(double dai, double rong)
{
    return dai * rong;
}

```

- Tiếp tục bổ sung khai báo nguyên mẫu hàm tính diện tích hình chữ nhật :

```
double DienTichHinhChuNhat(double dai, double rong);
```

6.2 Trong tập tin *menu.h* :

ta bổ sung xử lý thực hiện tính diện tích hình chữ nhật trong case 2 của hàm XuLyMenu. Trong hàm này ta khai báo thêm một biến thực b (đã có a và dienTich) để lưu trữ chiều rộng.

```

void XuLyMenu(int menu)
{
    // Khai báo các biến
    double a, b, dienTich;
    switch (menu)
    {
        // như 5.2
        case 2:
            cout << endl << "2. Tính diện tích hình chu nhật";

```

```
// Thông báo người dùng nhập chiều dài & rộng
cout << endl << "Nhập chiều dài hình CN : ";
cin >> a;

cout << endl << "Nhập chiều rộng hình CN : ";
cin >> b;

// Gọi hàm tính diện tích hình chữ nhật
dienTich = DienTichHinhChuNhat(a, b);

// Xuất kết quả
cout << endl << "Diện tích hình chữ nhật "
    << " có chiều dài = " << a
    << " chiều rộng = " << b << " là " << dienTich;
break;
// như 5.2
}
}
```

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có và kiểm tra việc thực hiện chức năng 2.

Bước 7 :

7.1 Trong *thuvien.h*,

Tiếp tục bổ sung :

- Trong phần định nghĩa nghĩa các hàm xử lý, ta định nghĩa hàm tính diện tích hình tam giác :

```
// Định nghĩa hàm tính diện tích hình tam giác
// Input : a = Độ dài cạnh a
//         b = Độ dài cạnh b
//         c = Độ dài cạnh c
// Output: Diện tích hình tam giác
double DienTichTamGiac(double a, double b, double c)
{
    double p = (a + b + c)/2;
    return sqrt(p * (p - a) * (p - b) * (p - c));
}
```

- Tiếp tục bổ sung khai báo nguyên mẫu hàm tính diện tích hình tam giác :

```
double DienTichTamGiac(double a, double b, double c);
```

7.2 Trong tập tin *menu.h* :

Ta bổ sung xử lý thực hiện tính diện tích hình tam giác trong case 3 của hàm XuLyMenu. Trong hàm này ta khai báo thêm một biến thực c (đã có a, b và dienTich) . a, b, c lưu trữ giá trị 3 cạnh tam giác.

```
void XuLyMenu(int menu)
{
    // Khai báo các biến
    double a, b, c, dienTich;
    switch (menu)
    {
```

//Như 6.2

case 3:

```
cout << endl << "3. Tính diện tích hình tam giác";
// Thông báo người dùng nhập chiều dài 3 cạnh
cout << endl << "Nhập độ dài cạnh a : ";
cin >> a;

cout << endl << "Nhập độ dài cạnh b : ";
cin >> b;

cout << endl << "Nhập độ dài cạnh c : ";
cin >> c;

// Gọi hàm tính diện tích hình tam giác
dienTich = DienTichTamGiac(a, b, c);

// Xuất kết quả
cout << endl << "Diện tích hình tam giác "
    << " có 3 cạnh a = " << a << ", b = " << b
    << ", c = " << c << " là " << dienTich;

break;
```

//như 6.2

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có và kiểm tra việc thực hiện chức năng 3.

Bước 8 :

8.1 Trong *thuvien.h*,

Tiếp tục bổ sung :

- Trong phần định nghĩa hằng, ta bổ sung định nghĩa hằng PI :

```
// =====
// Định nghĩa các hằng số và kiểu dữ liệu mới
// =====
#define PI 3.1415926
```

- Trong phần định nghĩa các hàm xử lý, ta định nghĩa hàm tính diện tích hình tròn :

```
// Định nghĩa hàm tính diện tích hình tròn
// Input : banKinh = Độ dài bán kính hình tròn.
// Output: Diện tích hình tròn
double DienTichHinhTron(double banKinh)
{
    return PI * banKinh * banKinh;
}
```

- Tiếp tục bổ sung khai báo nguyên mẫu hàm tính diện tích hình tròn :

```
double DienTichHinhTron(double banKinh);
```

8.2 Trong tập tin **menu.h** : ta bổ sung xử lý thực hiện tính diện tích hình tròn trong case 4 của hàm XuLyMenu. Trong hàm này ta khai báo thêm biến thực r để lưu trữ bán kính hình tròn nhập từ bàn phím. Hàm XuLyMenu đầy đủ như sau :

```
void XuLyMenu(int menu)
{
    // Khai báo các biến
    double a, b, c, r, dienTich;
    switch (menu)
    {
        case 0:
            cout << endl << "0. Thoat khoi chuong trinh";
            break;

        case 1:
            cout << endl << "1. Tinh dien tich hinh vuong";
            // Thông báo người dùng nhập độ dài cạnh
            cout << endl << "Nhap do dai canh hinh vuong : ";
            cin >> a;

            // Gọi hàm tính diện tích hình vuông
            dienTich = DienTichHinhVuong(a);

            // Xuất kết quả
            cout << endl << "Dien tich hinh vuong canh "
                << a << " la " << dienTich;
            break;

        case 2:
            cout << endl << "2. Tinh dien tich hinh chu nhat";
            // Thông báo người dùng nhập chiều dài & rộng
            cout << endl << "Nhap chieu dai hinh CN : ";
            cin >> a;

            cout << endl << "Nhap chieu rong hinh CN : ";
            cin >> b;

            // Gọi hàm tính diện tích hình chữ nhật
            dienTich = DienTichHinhChuNhat(a, b);

            // Xuất kết quả
            cout << endl << "Dien tich hinh chu nhat "
                << " co chieu dai = " << a
                << " chieu rong = " << b << " la " << dienTich;
            break;

        case 3:
            cout << endl << "3. Tinh dien tich hinh tam giac";
            // Thông báo người dùng nhập chiều dài 3 cạnh
            cout << endl << "Nhap do dai canh a : ";
```

```

cin >> a;

cout << endl << "Nhập độ dài cạnh b : ";
cin >> b;

cout << endl << "Nhập độ dài cạnh c : ";
cin >> c;

// Gọi hàm tính diện tích hình tam giác
dienTich = DienTichTamGiac(a, b, c);

// Xuất kết quả
cout << endl << "Diện tích hình tam giác "
    << " có 3 cạnh a = " << a << ", b = " << b
    << ", c = " << c << " là " << dienTich;
break;

case 4:
    cout << endl << "4. Tính diện tích hình tròn";
    // Thông báo người dùng nhập độ dài bán kính
    cout << endl << "Nhập độ dài bán kính : ";
    cin >> a;

    // Gọi hàm tính diện tích hình tròn
    dienTich = DienTichHinhTron(a);

    // Xuất kết quả
    cout << endl << "Diện tích hình tròn bán kính "
        << a << " là " << dienTich;
    break;
}
}

```

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.
 Kiểm tra việc thực hiện các chức năng.
 Kết thúc chương trình.

Bài 3: Chương trình tính tổng các dãy số

Trong phần này, ta sẽ xây dựng chương trình để tính tổng các dãy số sau:

$$\begin{aligned}
 \bullet \quad R &= \sum_{i=1}^n \frac{1}{i} & \bullet \quad S &= \sum_{i=1}^n \frac{i+1}{i^2} & \bullet \quad T &= \sum_{i=1}^n \frac{(-1)^i i}{i+1}
 \end{aligned}$$

Bước 1. Tạo dự án Win32 Console Application mới. Đặt tên là **Lab04_C_Bai3_TongDaySo**.

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

Bước 3: Trong tập tin **menu.h** ta viết lại như sau (cấu trúc giống như bước 9 mục 2, chỉ thay đổi nội dung theo yêu cầu bài toán) :


```
// =====
// 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

// Input : Không có

// Output: Không có

```
void XuatMenu()
{
    cout << endl << "===== HE THONG CHUC NANG =====";
    cout << endl << "0. Thoat khoi chuong trinh";
    cout << endl << "1. Tinh tong R";
    cout << endl << "2. Tinh tong S";
    cout << endl << "3. Tinh tong T";
    cout << endl << "===== ";
}
```

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 << "\nNhập 1 số không khoảng [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 :

Chú ý rằng cả 3 tổng này đều có đầu vào chung là số nguyên dương n (kích thước khi dùng của dãy số). Vì vậy ta bổ sung n làm đối của hàm **XuLyMenu** (ngoài tham số menu)

// Input : menu = Số thứ tự menu do người chọn.

// n = Một số nguyên dương

// Output: Không có.

```

void XuLyMenu(int menu, unsigned int n)
{
    switch (menu)
    {
        case 0:
            cout << endl << "0. Thoat khoi chuong trinh";
            break;
        case 1:
            cout << endl << "1. Tinh tong R";
            //bổ sung sau
            break;
        case 2:
            cout << endl << "2. Tinh tong S";
            //bổ sung sau
            break;
        case 3:
            cout << endl << "3. Tinh tong T";
            //bổ sung sau
            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, unsigned int n);

```

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

Trong bước tiếp theo, ta soạn thảo từng phần (chức năng) trong tập tin *thuvien.h*, bổ sung xử lý chức năng trong hàm *XuLyMenu* của *menu.h*, viết lại hàm *ChayChuongTrinh* trong *program.cpp* để điều khiển tùy chọn thực hiện menu chương trình (chọn 0 là dừng chương trình).

Bước 4 :

4.1 Trong tập tin program.cpp, ta cập nhật lại hàm ChayChuongTrinh :

```

// Định nghĩa hàm lặp lại việc chọn menu, xử lý menu
// cho tới khi người dùng chọn menu 0 thì thoát CT.
void ChayChuongTrinh()
{
    // Khai báo biến
    int menu,           // lưu số thứ tự menu được chọn
        soMenu = 3;    // lưu số lượng menu chức năng.
    // Khai báo biến n
    unsigned int n;

    // Nhập giá trị cho biến n
    cout << endl << "Nhap mot so nguyen duong : ";
    cin >> n;
    // Lặp lại việc chọn và xử lý menu cho tới khi

```

```
// người dùng chọn chức năng 0. Thoát khỏi CT.
do
{
    menu = ChonMenu(soMenu);
    XuLyMenu(menu,n);
} while (menu > 0);
}
```

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

Kiểm tra thực hiện chức năng chọn 0 thì dừng chương trình, chọn từ 1 đến 3 thì không làm gì cả.

Bước 5:

5.1 Trong *thuvien.h*,

Bổ sung:

- Trong phần định nghĩa nghĩa các hàm xử lý, ta định nghĩa hàm tính TongR :

```
// Định nghĩa hàm tính tổng  $1 + 1/2 + 1/3 + \dots + 1/n$ 
// Input : n = một số nguyên dương.
// Output: Tổng  $1 + 1/2 + 1/3 + \dots + 1/n$ 
double TinhTongR(unsigned int n)
{
    double sum = 0;           // Khai báo biến lưu kết quả
    unsigned int i;
    // Duyệt qua từng số i từ 1 đến n
    for ( i=1; i<=n; i++)
        sum += 1.0 / i; // Cộng dồn 1/i vào kết quả

    return sum;
}
```

- Trong phần khai báo nguyên mẫu hàm, bổ sung khai báo nguyên mẫu hàm tính TongR :

```
double TinhTongR(unsigned int n);
```

5.2 Trong tập tin *menu.h* :

Ta bổ sung thực hiện xử lý tính TinhTongR .

Trong hàm nay, ta khai báo biến thực sum để lưu trữ giá trị tổng cần tính của dãy số.

```
void XuLyMenu(int menu, unsigned int n)
{
    double sum;
    switch (menu)
    {
        case 0:
            cout << endl << "0. Thoat khoi chuong trinh";
            break;
        case 1:
            cout << endl << "1. Tinh tong R";
            // Gõi hàm tính tổng R
            sum = TinhTongR(n);
    }
}
```

```

        // Xuất kết quả
        cout << endl << "R = " << sum;
        break;
    case 2:
        cout << endl << "2. Tính tong S";
        //bổ sung sau
        break;
    case 3:
        cout << endl << "3. Tính tong T";
        //bổ sung sau
        break;
    }
    _getch();
}

```

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

Tiếp theo bổ sung các chức năng khác.

Bước 6 :

5.1 Trong *thuvien.h*,

Ta bổ sung:

- Trong phần định nghĩa nghĩa các hàm xử lý, ta định nghĩa hàm tính TinhTongS :

// Định nghĩa hàm tính tổng $2/1 + 3/4 + 4/9 + \dots$

// Input : n = một số nguyên dương.

// Output: Tổng $2/1 + 3/4 + 4/9 + \dots + (n+1)/(n^2)$

double TinhTongS(unsigned int n)

{

double sum = 0; // Khai báo biến lưu kết quả
unsigned int i;

// Duyệt qua từng số i từ 1 đến n
// và cộng dồn $(i+1)/(i^2)$ vào kết quả

for (i=1; i<=n; i++)
sum += (i + 1.0) / (i * i);

return sum;

}

- Trong phần khai báo nguyên mẫu hàm, bổ sung khai báo nguyên mẫu hàm tính TinhTongS :

double TinhTongS(unsigned int n);

6.2 Trong tập tin *menu.h* : ta bổ sung thực hiện xử lý tính TongS (trong case 2:)

void XuLyMenu(int menu, unsigned int n)

{

double sum = 0;

switch (menu)

{

case 0:

```

        cout << endl << "0. Thoat khỏi chương trình";
        break;
    case 1:
        cout << endl << "1. Tính tổng R";
        // Gọi hàm tính tổng R
        sum = TinhTongR(n);
        // Xuất kết quả
        cout << endl << "R = " << sum;
        break;
    case 2:
        cout << endl << "2. Tính tổng S";
        // Gọi hàm tính tổng S
        sum = TinhTongS(n);
        // Xuất kết quả
        cout << endl << "S = " << sum;
        break;
    case 3:
        cout << endl << "3. Tính tổng T";
        //bổ sung sau
        break;
    }
    _getch();
}

```

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

Kiểm tra việc thực hiện chức năng 2

Bổ sung các chức năng còn lại.

Bước 7 :

7.1 Trong *thuvien.h*, ta bổ sung:

- Trong phần định nghĩa nghĩa các hàm xử lý, ta định nghĩa hàm tính TinhTongT :

/ Định nghĩa hàm tính tổng $-1/2 + 2/3 - 3/4 + \dots$

// Input : n = một số nguyên dương.

// Output: Tổng $-1/2 + 2/3 - 3/4 + \dots + n * (-1)^n / (n+1)$

```

double TinhTongT(unsigned int n)
{
    double sum = 0;           // Khai báo biến lưu kết quả
    unsigned int i;
    // Duyệt qua từng số i từ 1 đến n và cộng dồn
    // hoặc trừ bớt  $i * (-1)^i / (i+1)$  khỏi kết quả
    for (i=1; i<=n; i++)
        if (i % 2 == 0)
            sum += i / (i + 1.0);
        else
            sum -= i / (i + 1.0);

    return sum;
}

```

- Trong phần khai báo nguyên mẫu hàm, bổ sung khai báo nguyên mẫu hàm tính TinhTongT :

`double TinhTongT(unsigned int n);`

7.2 Trong tập tin **menu.h** : ta bổ sung thực hiện xử lý tính TinhTongT (trong case 3:)

```
void XuLyMenu(int menu, unsigned int n)
{
    switch (menu)
    {
        case 0:
            cout << endl << "0. Thoat khoi chuong trinh";
            break;
        case 1:
            cout << endl << "1. Tinh tong R";
            // Gọi hàm tính tổng R
            sum = TinhTongR(n);
            // Xuất kết quả
            cout << endl << "R = " << sum;
            break;
        case 2:
            cout << endl << "2. Tinh tong S";
            // Gọi hàm tính tổng S
            sum = TinhTongS(n);
            // Xuất kết quả
            cout << endl << "S = " << sum;
            break;
        case 3:
            cout << endl << "3. Tinh tong T";
            // Gọi hàm tính tổng T
            sum = TinhTongT(n);
            // Xuất kết quả
            cout << endl << "T = " << sum;
            break;
    }
    _getch();
}
```

Nhấn Ctrl+F5 để chạy chương trình, sửa lỗi nếu có.
Kiểm tra việc thực hiện chức năng.
Kết thúc chương trình.

D. Bài tập bắt buộc

Các bài tập dưới đây yêu cầu phải định nghĩa hàm, tổ chức menu và tạo cấu trúc chương trình như đã chỉ dẫn trong mục 1 của phần C. Tên hàm, hằng và biến phải được đặt tên theo đúng quy ước như trong Lab 3.

1. Số nguyên

Viết chương trình cho phép người dùng nhập vào một số nguyên dương n . Sau đó:

- Xuất các số từ 1 tới n , các số cách nhau 1 dấu Tab và mỗi dòng chứa 10 số.
- Đếm số lượng các số chia hết cho 3 nhưng không chia hết cho 4 trong đoạn $[1..n]$.

- c. Đếm số lượng chữ số của n . Ví dụ: $n = 12345$ thì số lượng chữ số là 5.
- d. Đảo ngược số n . Ví dụ: $n = 12345$ thì sau khi đảo ngược, $n = 54321$.
- e. Tính tổng các chữ số trong n . Ví dụ: $n = 12345$ thì tổng các chữ số là $1+2+3+4+5 = 15$.
- f. Cho biết chữ số đầu tiên trong n . Ví dụ: $n = 12345$ thì chữ số đầu tiên là 1.
- g. Đổi số n sang hệ nhị phân. Ví dụ: $n = 1234_{10}$ tương ứng với 10011010010_2 .
- h. Kiểm tra số n có phải là số hoàn hảo (còn gọi là số hoàn chỉnh, số hoàn thiện)? Biết rằng, số hoàn hảo là số có tổng các ước bằng hai lần chính nó. Ví dụ: 6, 28 là số hoàn chỉnh vì $6 = (1+2+3+6)/2$, $28 = (1+2+4+7+14+28)/2$.
- i. Xuất tất cả các số hoàn chỉnh trong phạm vi $[1..n]$. Ví dụ: Với $n = 30$ thì xuất 6, 28.
- j. Tìm số nguyên m lớn nhất sao cho tổng $1+2+\dots+m \leq n$.

2. Đổi cơ số

Viết chương trình cho phép người dùng nhập vào số nguyên dương n . Sau đó, xuất ra màn hình số n ở hệ cơ số b ($2 \leq b \leq 16$) do người dùng chọn. Trường hợp $10 \leq b \leq 16$, sử dụng các ký tự A đến F để biểu diễn các số từ 10 tới 16. Chương trình phải hiển thị các menu sau:

- a. Đổi sang hệ nhị phân ($b=2$)
- b. Đổi sang hệ bát phân ($b=8$)
- c. Đổi sang hệ thập lục phân ($b=16$)
- d. Đổi sang hệ cơ số 7 ($b=7$)

3. Số nguyên tố

Viết chương trình cho phép người dùng nhập vào một số nguyên dương n . Sau đó:

- a. Kiểm tra n có phải là số nguyên tố (sử dụng vòng lặp **for**).
- b. Xuất ra màn hình các số nguyên tố trong phạm vi từ 1 tới n , các số cách nhau 1 dấu Tab. Mỗi dòng chứa 5 số.
- c. Đếm số lượng số nguyên tố trong phạm vi từ 1 tới n .
- d. Tính tổng các ước số nguyên tố của n . Ví dụ: $n = 30$ có các ước 1, 2, 3, 5, 6, 10, 15, 30. Tổng các ước số nguyên tố là $2 + 3 + 5 = 10$.
- e. Phân tích n thành tích các thừa số nguyên tố. Ví dụ: $n = 12 = 2.2.3$, $n = 30 = 2.3.5$.

4. Ước số

Viết chương trình cho phép người dùng nhập vào một số nguyên dương n . Sau đó:

- a. Xuất tất cả các ước số của n . Ví dụ: Các ước số của $n = 12$ là 1, 2, 3, 4, 6, 12.
- b. Đếm số lượng các ước số của n . Ví dụ: Số lượng ước số của $n = 12$ là 6.
- c. Tính tổng các ước số của n . Ví dụ: Tổng các ước số của $n = 12$ là 28.
- d. Tính căn bậc 2 của n (không sử dụng hàm **sqrt**).
- e. Tìm số lớn nhất nhỏ hơn hoặc bằng n mà là lũy thừa của 2. Ví dụ: $n = 1234$ thì xuất 1024.

5. Khai triển Taylor

Viết chương trình tính giá trị các hàm $\sin(x)$, $\cos(x)$, e^x với giá trị x được nhập từ bàn phím. Yêu cầu: không được sử dụng các hàm **sin**, **cos** trong thư viện **math.h**.

Hướng dẫn: Sử dụng các công thức khai triển Taylor sau với độ chính xác 10^{-5}

$$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots, -\infty < x < \infty$$

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} - \frac{x^{11}}{11!} + \dots$$

$$\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \frac{x^8}{8!} - \frac{x^{10}}{10!} + \dots$$

6. Số Fibonacci

Dãy Fibonacci là dãy vô hạn các số tự nhiên bắt đầu bằng hai phần tử 0 và 1, các phần tử sau đó được thiết lập theo quy tắc mỗi phần tử luôn bằng tổng hai phần tử trước nó cộng lại. Công thức truy hồi của dãy Fibonacci là:

$$F(n) = \begin{cases} 0 & \text{khi } n = 0 \\ 1 & \text{khi } n = 1 \\ F(n-1) + F(n-2) & \text{khi } n \geq 2 \end{cases}$$

Viết chương trình cho phép người dùng nhập vào một số nguyên dương n . Sau đó:

- Tìm và xuất ra số Fibonacci thứ n .
- Liệt kê các số Fibonacci nhỏ hơn hoặc bằng n .
- Liệt kê n số Fibonacci đầu tiên.

E. Bài tập làm thêm

1. Giải phương trình

Viết chương trình giải các phương trình và hệ phương trình sau:

- Phương trình bậc nhất 1 ẩn
- Phương trình bậc hai 1 ẩn
- Phương trình bậc ba 1 ẩn
- Phương trình trùng phương
- Hệ phương trình bậc nhất 1 ẩn
- Hệ phương trình bậc nhất 2 ẩn

2. Dãy số

Viết chương trình tính các tổng sau:

- $H_n = \sum_{i=1}^n \frac{1}{i} = 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}$ (còn gọi là các số Harmonic)
- $S_n = \sum_{i=1}^n \frac{i+1}{i^2} = 2 + \frac{3}{4} + \frac{4}{9} + \dots + \frac{n+1}{n^2}$
- $T_n = \sum_{i=1}^n \frac{(-1)^i i}{i+1} = -\frac{1}{2} + \frac{2}{3} - \frac{3}{4} + \dots + \frac{(-1)^n n}{n+1}$
- $U_n = \sum_{i=1}^n \frac{1}{i(i+1)} = \frac{1}{1.2} + \frac{1}{2.3} + \frac{1}{3.4} + \dots + \frac{1}{n(n+1)}$
- $F_n = \sum_{i=1}^n i! = 1! + 2! + 3! + \dots + n!$

3. Hình học 3 chiều

Viết chương trình tính thể tích của các hình sau:

- Hình lập phương
- Hình hộp chữ nhật
- Hình cầu
- Hình nón
- Hình trụ tròn
- Hình chóp

Tham khảo cách tính thể tích các hình 3D tại: <http://www.mathvn.com/2014/09/cong-thuc-tinh-tich-khoi-chop-lang-tru.html>

