

BÀI THỰC HÀNH SỐ 1: SỬ DỤNG C FREE

Mục Tiêu

- 1/. Sinh viên tìm hiểu, khai thác và sử dụng Turbo C, C-Free, Dev C++, ... để hiện thực hóa các chương trình được viết bằng Ngôn ngữ lập trình C.
- 2/. Viết một số chương trình C đơn giản, dịch chương trình nguồn và xử lý lỗi cú pháp, chạy chương trình và kiểm tra kết quả (ngữ nghĩa).

I/. KHAI THÁC CÔNG CỤ

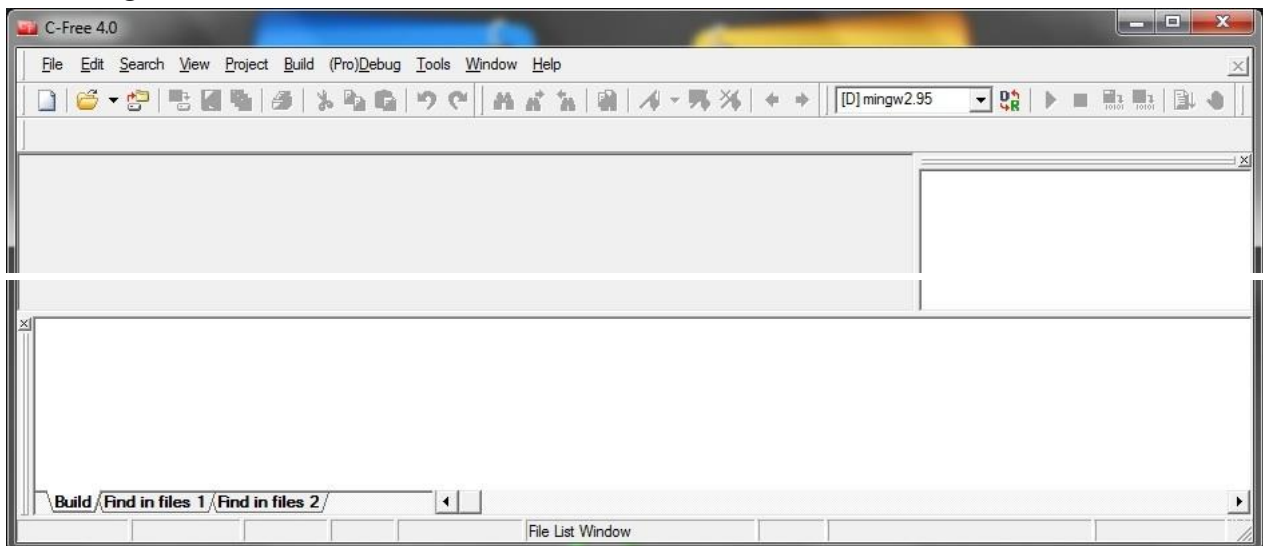
1. Cài đặt

Để bắt đầu cho việc học lập trình bằng ngôn ngữ C, sinh viên cần cài đặt một trong các phần mềm Turbo C, C-Free, Dev C ++, ... trên máy tính. Thực tế, các phòng máy của IUH đã cài đặt sẵn C Free, vì vậy chúng ta thống nhất sẽ sử dụng phần mềm này trong suốt quá trình thực hành học phần Phương pháp lập trình.

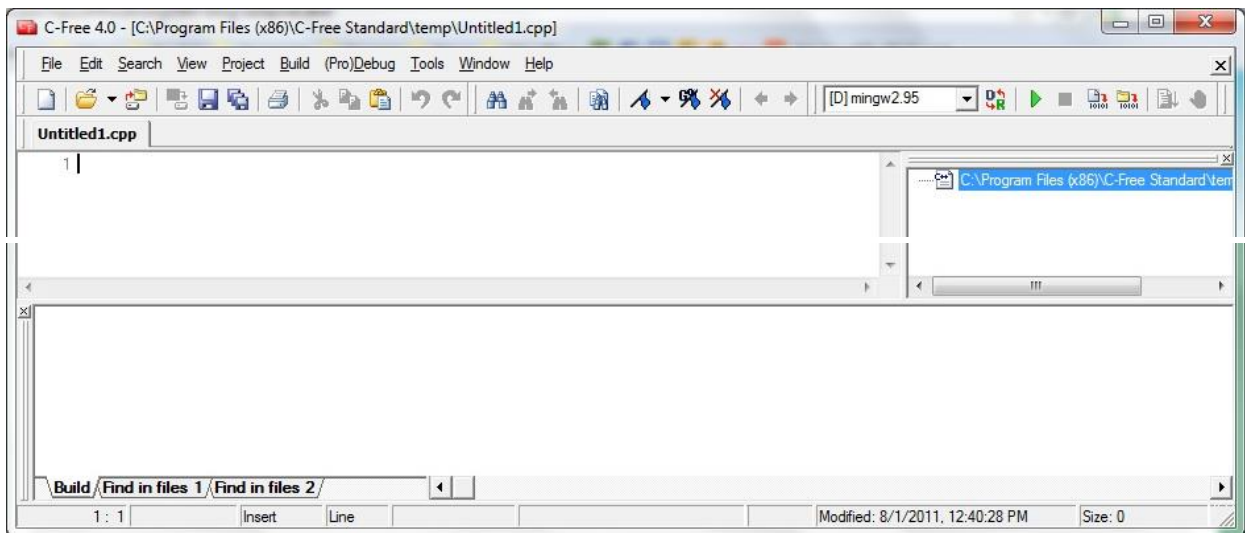
Đối với PC, Laptop cá nhân, sinh viên Google với keyword **C-Free Download** để download C-Free phiên bản 4.0, sau đó tiến hành setup theo hướng dẫn của phần mềm.

2. Sử dụng Editor của C-Free

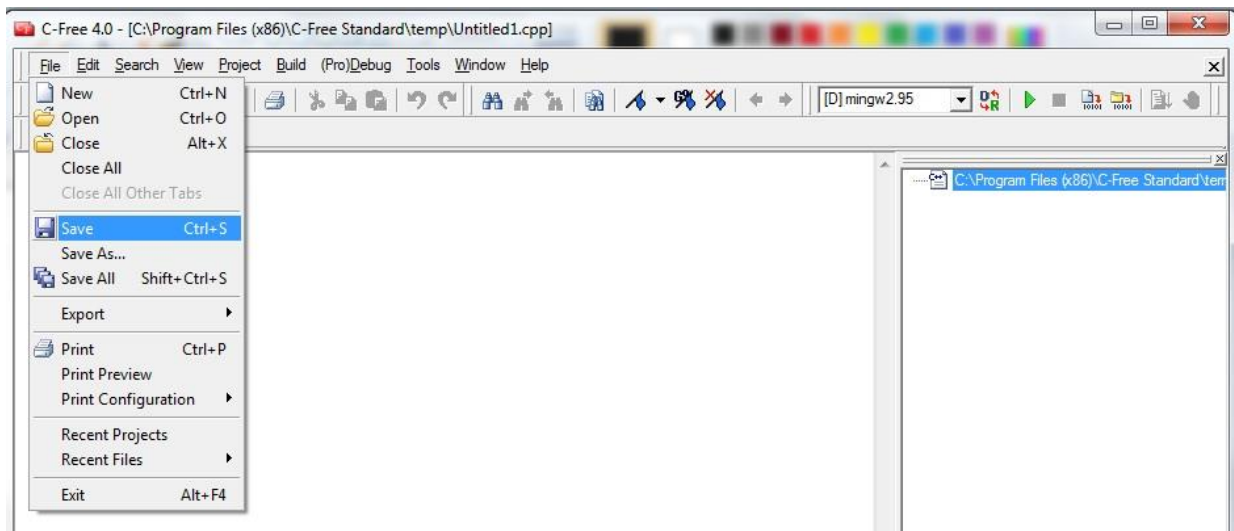
- Khởi động C-Free, màn hình xuất hiện:



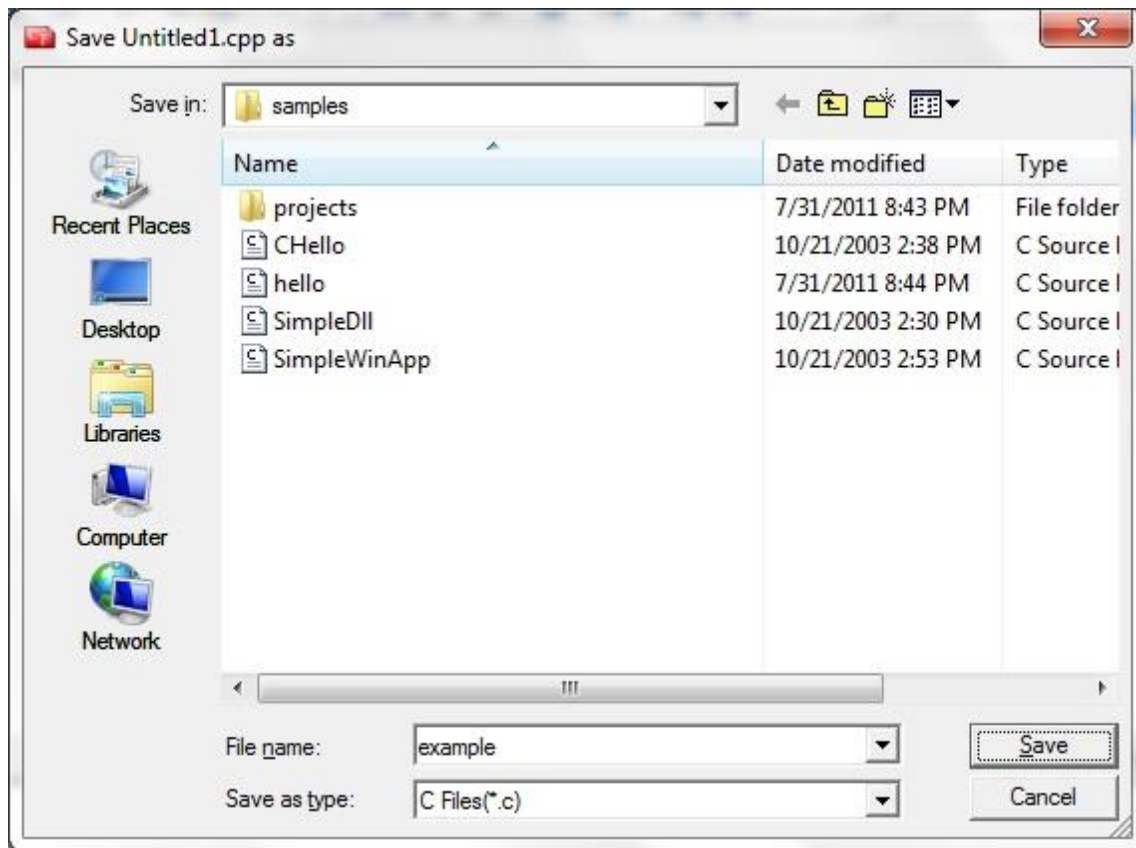
- **Tạo file mới:** Vào menu File, → Chọn New (hoặc sử dụng bàn phím bằng cách bấm tổ hợp phím Ctrl+N). Lúc này một file tên Untitled1.cpp được tạo và dấu nhắc chuột được đặt trong ô soạn thảo (editor). Từ đây người dùng bắt đầu soạn thảo chương trình bằng Ngôn ngữ C.



- *Lưu File*: Trên menu File → Chọn Save (hoặc sử dụng tổ hợp phím Ctrl + S).



Tiếp theo, chọn vị trí (không gian) để lưu code chương trình bằng việc sử dụng browser để thay đổi thư mục chứa tập tin là code của chương trình ở *Save in*. Trong ô *Save as type* chọn *C Files (*.c)* và trong ô *File name* chọn ghi tên chương trình. Ví dụ như trong hình Cfree sẽ thay đổi tên và đuôi của file. *Untitled1.cpp* chuyển thành *example.c*. Các bạn cũng có thể giữ nguyên phần mở rộng của tập tin là **cpp**.



- Quan sát title của editor, tên tập tin đã thay đổi thành example.c.



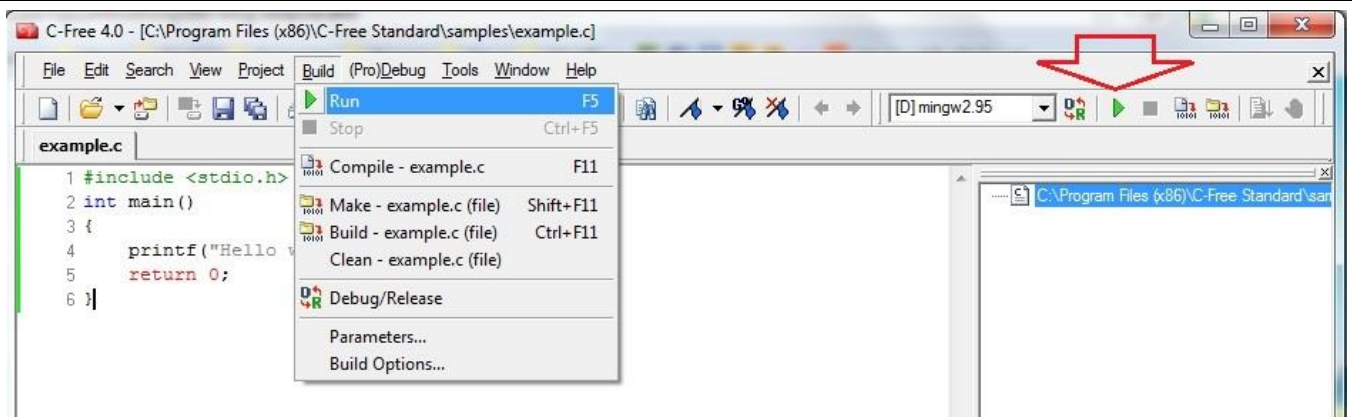
3. Dịch và Chạy chương trình.

Sau khi hoàn thành các bước ở 2. Bây giờ bạn có thể gõ code chương trình được viết bằng Ngôn ngữ C. Hãy bắt đầu với một chương trình đơn giản: In ra dòng chữ Hello world.

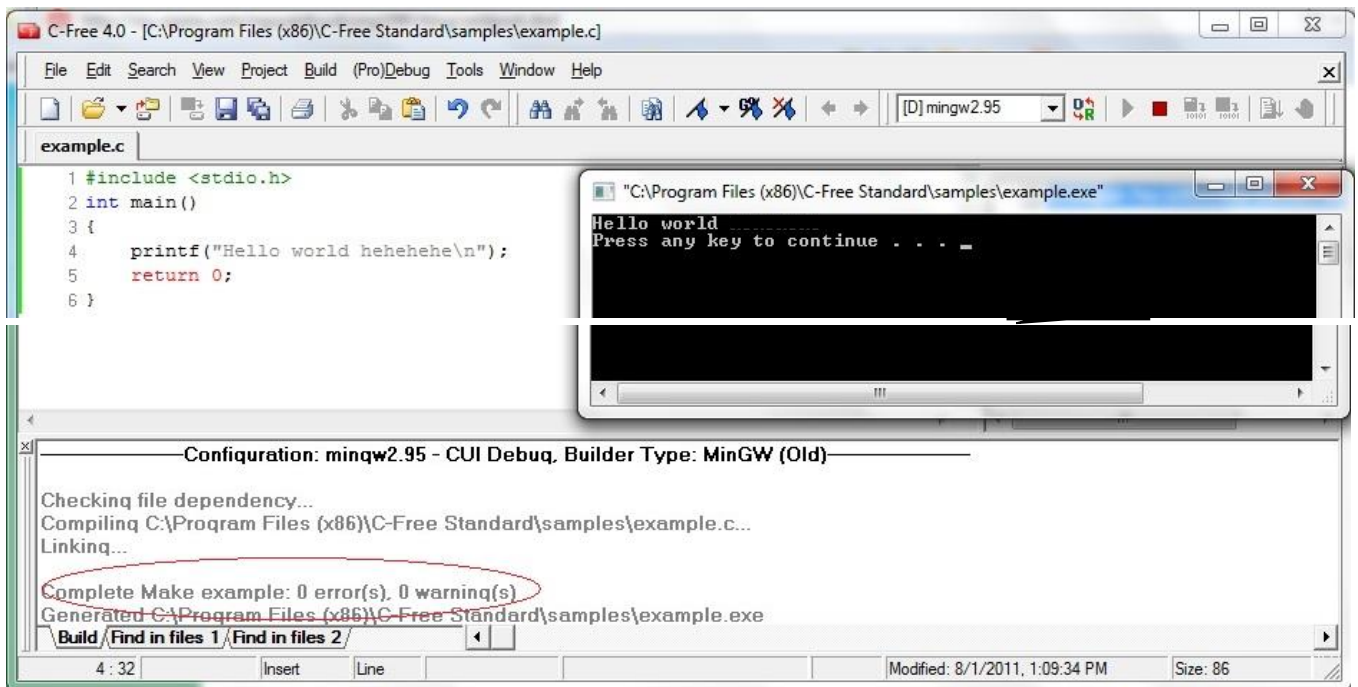
Soạn thảo: Trong ô soạn thảo, hãy gõ đoạn lệnh sau (gọi là mã nguồn, thuật ngữ tiếng Anh là code):

```
#include <stdio.h>
int main()
{
    printf("Hello world");
    return 0;
}
```

Chạy chương trình: Chạy chương trình để xem chương trình đã thực hiện đúng với ý muốn của người viết chương trình hay không. Trên menu Build → chọn Run (hoặc click chuột vào nút có hình tam giác màu xanh trên thanh công cụ).



Một cửa sổ màu đen hiện ra:



Các bạn chú ý đến phần khoanh đỏ. “**Complete Make exmple: 0 error(s), 0 warning(s)**” cho biết chương trình của ta không có lỗi, và đã biên dịch thành công. Trong cửa sổ đen hiện ra dòng chữ "Hello world". Chính là kết quả thực hiện chương trình. Trong trường hợp các bạn có sai/sót ở bất kỳ chỗ nào trong đoạn code, thông báo lỗi sẽ xảy ra. Lúc này bạn cần quay trở lại chương trình để sửa lỗi!!

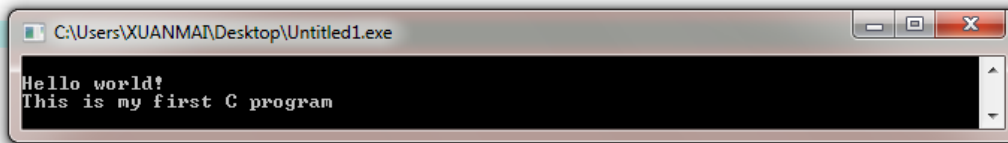
II. VIẾT MỘT SỐ CHƯƠNG TRÌNH ĐƠN GIẢN

4. Chương trình mẫu

Đừng buồn khoăn nếu các bạn chưa hiểu các câu lệnh cũng như tại sao phải viết như vậy. Mục đích chúng ta ở bài thực hành này là học cách vận hành công cụ C-Free để soạn thảo code và chạy thử các ví dụ có sẵn. Các câu lệnh sẽ được giải thích cụ thể trong các chương tiếp theo.

Ví dụ 1: Viết chương trình in ra màn hình 2 dòng văn bản:

```
#include <stdio.h>
#include <conio.h>
int main()
{
    printf("\nHello world!");
    printf("\nThis is my first C program");
    getch();
    return 0;
}
```



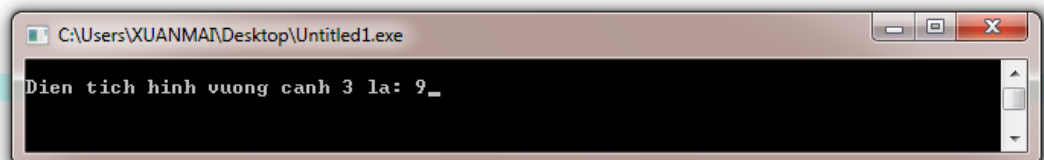
Trong ví dụ này các bạn thấy xuất hiện cặp ký tự `\n`, đây là *ký tự điều khiển* việc xuống dòng, để đưa con trỏ về đầu một dòng mới. Chương trình trên có thể được viết lại theo cách sau và cho ra cùng kết quả.

```
#include <stdio.h>
#include <conio.h>
int main()
{
    printf("\nHello world! \nThis is my first C program");
    getch();
    return 0;
}
```



Ví dụ 2: Viết chương trình tính diện tích hình vuông có cạnh a bằng 3.

```
#include <stdio.h>
#include <conio.h>
int main()
{
    int a=3;
    printf("\nDiện tích hình vuông cạnh %d là: %d", a, a*a);
    getch();
    return 0;
}
```



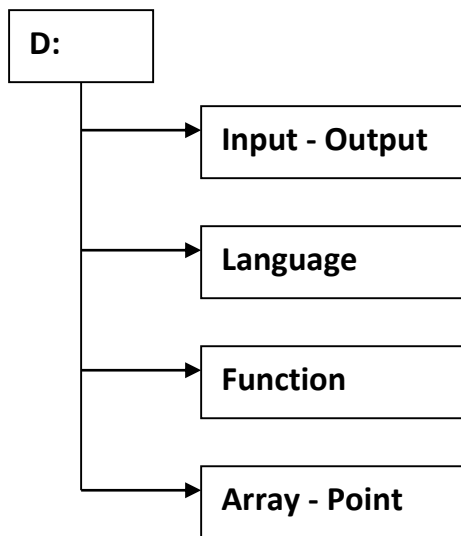
Ở ví dụ này, `%d` trong thân câu lệnh `printf` là *ký tự định dạng* để yêu cầu ghi các số a và `a*a` ra màn hình ở dạng số nguyên. p

5. Thực hành viết chương trình.

1. Khảo sát các chức năng trong Menu của C-Free: File, Edit, Serch, View, Build
2. Viết chương trình in ra màn hình một số dòng văn bản bất kỳ.
3. Viết chương trình tính diện tích hình tròn bán kính r, hình chữ nhật cạnh a và b.
4. Tìm hiểu câu lệnh `scanf` ở dạng căn bản (đã giới thiệu ở slides) để viết chương trình nhập vào 2 số nguyên a và b sau đó thông báo lên màn hình tổng, hiệu, tích và thương của chúng.

BÀI THỰC HÀNH SỐ 2: NHẬP XUẤT

1. Khởi động C Free.
2. Sử dụng Windows explore để tạo cây thư mục có cấu trúc như sau:



3. Soạn thảo chương trình dưới đây trong editor của Turbo C và lưu vào thư mục **Input-Output** với tên file *Test.CPP*

```
#include <stdio.h>
#include<conio.h>
int main ()
{
    int namsinh;
    printf("\nNhap vao sinh nam cua ban:");
    scanf("%d", &namsinh);
    printf("Tuoi cua ban la:%d", 2016 - namsinh);
    getch();
    return 0;
}
```

4. Cho thực hiện chương trình Test.CPP. Sau đó thêm các ký tự **//** trước dòng **#include<conio.h>** rồi thực hiện lại chương trình Test.CPP, quan sát và sửa lỗi để chương trình thực hiện được.
5. Để kéo dài “hạn sử dụng” cho chương trình tính tuổi, hãy nhập vào năm hiện tại sau đó tính tuổi và in kết quả ra màn hình.
6. Viết chương trình tính diện tích và chu vi của hình tròn có độ dài bán kính *r* nhập từ bàn phím. Lưu vào thư mục **Input-Output** với tên file *Circle.CPP*

7. Viết chương trình tính diện tích và chu vi của hình chữ nhật có chiều dài a và chiều rộng b nhập từ bàn phím. Lưu vào thư mục **Input-Output** với tên file *Rectangle.CPP*
8. Viết chương trình nhập vào số ngày ở khách sạn của một khách hàng sau đó thông báo lên màn hình số tiền mà họ phải trả (Lưu vào thư mục **Input-Output** với tên file *Service.CPP*). Biết rằng công thức tính tiền của khách sạn như sau:

$$[\text{Tiền trả}] = \text{<số tuần>} * 700000 + \text{<số ngày>} * 120000.$$

Trong đó <số ngày> là những ngày dư ra, không nằm trong tuần. Ví dụ: Nếu số ngày lưu trú ở khách sạn là 12 ngày thì sẽ tương đương 1 tuần + 5 ngày.

9. Viết chương trình tính giá trị của biểu thức $P = \frac{a^3 - b^2 - 4}{a^2 + b^2 + 1}$, với a và b nhập từ bàn phím. Lưu vào thư mục **Input-Output** với tên file *Expression.CPP*

10. Viết chương trình nhập vào số ngày công làm việc trong tháng của một nhân viên sau đó thông báo lên màn hình số tiền lương mà họ nhận được (Lưu vào thư mục **Input-Output** với tên file *Salary.CPP*). Biết rằng công thức tính lương của công ty như sau:

$$[\text{Lương}] = \text{<số ngày công>} * 300.000$$

Trong đó: Nếu <số ngày công> lớn hơn 22 thì những ngày dư ra (là việc ngoài giờ) được tính gấp đôi.

11. Viết chương trình nhập vào một số thể hiện $^{\circ}\text{C}$ sau đó in ra màn hình $^{\circ}\text{F}$ tương ứng. Lưu vào thư mục **Input-Output** với tên file *De.CPP*
12. Viết chương trình nhận vào 3 số nguyên lần lượt tương ứng với h:m:s của một đồng hồ điện tử. Hỏi sau 1 giây, màn hình đồng hồ hiển thị như thế nào? Lưu vào thư mục **Input-Output** với tên file *Clock.CPP*
13. Viết chương trình nhận vào 4 thực. Hiển thị lên màn số lớn nhất trong 4 số đó. Lưu vào thư mục **Input-Output** với tên file *Max.CPP*
14. Viết chương trình nhận vào 4 số thực. Hiển thị lên màn số bé nhất trong 4 số đó. Lưu vào thư mục **Input-Output** với tên file *Min.CPP*
15. Viết chương trình nhập vào 3 số thực a, b, c, sau đó kiểm tra chúng có phải 3 cạnh của một tam giác hay không? Lưu vào thư mục **Input-Output** với tên file *TestA.CPP*
16. Viết chương trình nhập vào 3 số nguyên a, b, c, sau đó kiểm tra chúng có phải 3 cạnh của một tam giác vuông hay không? Lưu vào thư mục **Input-Output** với tên file *TestB.CPP*
17. Viết chương trình nhập vào 3 số thực a, b, c, sau đó kiểm tra chúng có phải 3 cạnh của một tam giác đều hay không? Lưu vào thư mục **Input-Output** với tên file *TestC.CPP*
18. Viết chương trình nhập vào 2 cặp tọa độ (x_1, y_1) và (x_2, y_2) tương ứng với 2 điểm A và B trên mặt phẳng. Tính và hiển thị lên màn hình độ dài của đoạn thẳng AB. Lưu vào thư mục **Input-Output** với tên file *AB.CPP*

19. Viết chương trình nhập vào bán kính r của một hình tròn có tâm ở tọa độ (x_1, y_1) sau đó nhập vào tọa độ (x_2, y_2) của điểm A. Thông báo lên màn hình A ở trong, trên hay ngoài hình tròn. Lưu vào thư mục **Input-Output** với tên file *Circle.CPP*
20. Viết chương trình nhập vào số bài tập bạn đã hoàn thành sau đó hiển thị lên màn hình “Well done!” nếu số bài tập hoàn thành từ 10 trở lên (kể cả bài tập số 20) và “Try more!!” nếu số bài tập hoàn thành nhỏ hơn 10. Lưu vào thư mục **Input-Output** với tên file *Final.CPP*

BÀI THỰC HÀNH SỐ 3: CẤU TRÚC ĐIỀU KHIỂN, CẤU TRÚC LẶP**Mục đích:**

- Sử dụng các câu lệnh điều khiển
- Sử dụng các câu lệnh vòng lặp

Bài 01: Giải phương trình bậc hai. Tên chương trình: *SolQuad.cpp*

Hãy tính nghiệm của phương trình bậc hai:

$$ax^2 + bx + c = 0$$

Dữ liệu vào từ màn hình:

- Nhập ba số nguyên a, b, c từ màn hình ($a \neq 0$).

Kết quả ra màn hình:

Có các trường hợp sau:

- **Phương trình có hai nghiệm phân biệt:** in ra hai nghiệm theo thứ tự từ nhỏ đến lớn. Ghi với độ chính xác là hai chữ số ở hàng thập phân.
- **Phương trình có nghiệm duy nhất:** in ra nghiệm kép của phương trình, ghi với độ chính xác là hai chữ số ở hàng thập phân.
- **Phương trình vô nghiệm:** in ra từ 'Phuong trinh VO NGHIEM'.

Dữ liệu vào	Kết quả tương ứng
a = 1 b = -4 c = 3	Phuong trinh co hai nghiem: x1 = 1.00 x2 = 3.00
a = 1 b = -2 c = 1	Phuong trinh co nghiem kep: x1 = x2 = 1.00
a = 1 b = 2 c = 2	Phuong trinh VO NGHIEM.

Mở rộng: Giải phương trình bậc 2: $ax^2 + bx + c = 0$ với các hệ số a, b, c bất kỳ được nhập từ bàn phím.

- Lưu file mới (save as) **E-SolQuad.cpp**

- Gợi ý thuật toán:

Nếu ($a \neq 0$) giải phương trình bậc 2 (như vừa thực hiện ở trên)

Ngược lại, giải phương trình bậc nhất dạng $bx + c = 0$

- Test chương trình với 3 bộ dữ liệu ở trên và thêm các trường hợp (0, 2, 3); (0, 0, 3); (0, 0, 0).

Bài 02: Đọc chữ số. Tên chương trình: *Digit.cpp*

Hãy viết chương trình nhập một chữ số từ màn hình, ghi chữ số đó dưới dạng chữ

Dữ liệu vào từ màn hình:

- Đọc các chữ số đến khi nào chữ số nhập vào nằm ngoài khoảng 0..9

Kết quả ra màn hình:

- Tương ứng mỗi chữ số từ 0..9 ghi ra màn hình dạng chữ của nó.

Dữ liệu vào	Kết quả tương ứng
digit = 0	chu so khong
digit = 1	chu so mot
digit = 9	chu so chin
digit = -1	

Bài 03: Đánh giá điểm thi theo hệ tín chỉ. Tên chương trình *Credit.cpp*

Hãy viết chương trình nhập vào điểm thi môn TIN1013 theo thang điểm 10 sau đó phiên sang thang điểm 4 và điểm chữ (A, B+, B, C+, C, D+, D, F) tương ứng (xem bảng sau)

Điểm thang điểm 4	Điểm chữ	Điểm thang điểm 4	Điểm chữ
mark = 4.00	A	$2.00 \leq \text{mark} < 2.50$	C
$3.50 \leq \text{mark} < 4.00$	B+	$1.50 \leq \text{mark} < 2.00$	D +
$3.00 \leq \text{mark} < 3.50$	B	$1.00 \leq \text{mark} < 1.50$	D
$2.50 \leq \text{mark} < 3.00$	C+	$\text{mark} < 1.00$	F

Dữ liệu vào từ màn hình:

- Đọc vào điểm thi nằm trong khoảng 0..10 (số thực)

Kết quả ra màn hình:

- Tương ứng với mỗi điểm số (mark) nhập từ bàn phím nằm trong [0..10], phiên sang điểm 4 và cho biết điểm chữ tương ứng.
- Trường hợp điểm số (mark) nhỏ hơn 0 hoặc lớn hơn 10 thì in ra **Khong hop le**

Dữ liệu vào	Kết quả tương ứng
mark = 0	Diem theo thang 4 la 0.00 – Diem F
mark = 5	Diem theo thang 4 la 2.00 – Diem C
mark = 7.5	Diem theo thang 4 la 3.00 – Diem B
mark = 8	Diem theo thang 4 la 3.20 – Diem B+

mark = 10	Diem theo thang 4 la 4.00 - Diem A
mark = -1	Khong hop le
mark = 11	Khong hop le

Mục đích: Sử dụng câu lệnh điều khiển rẽ nhánh – switch (miền giá trị là số nguyên)**Bài 01: Tên chương trình: `Shape.cpp`**

Lập chương trình đưa ra bảng chọn như sau:

Chương trình tính diện tích các hình cơ bản

1. Hình tam giác
2. Hình chữ nhật
3. Hình tròn
4. Thoát

Bạn hãy nhập một số để chọn hình cần tính (1, 2, 3, hoặc 4):

Sau khi người dùng chọn một số, chương trình phải tính diện tích hình đó (nếu nhập các số ngoài 1, 2, 3, 4 thì quay lại menu chọn):

1. Yêu cầu người dùng nhập độ dài 3 cạnh của 1 tam giác, kiểm tra 3 cạnh đó có phải là 3 cạnh của tam giác không, nếu đúng thì đưa ra diện tích tam giác đó nếu không thì in ra thông báo "Ba số không phải là 3 cạnh của một tam giác" và quay lại menu chọn. (Diện tích tam giác tính theo công thức Hê-rông)

a =
b =
c =
Diện tích: ...
2. Yêu cầu người dùng nhập chiều dài và chiều rộng của hình chữ nhật và đưa ra diện tích hình chữ nhật đó (giả sử người dùng đã nhập đúng).

a =
b =
Diện tích: ...
3. Yêu cầu người dùng nhập bán kính hình tròn và tính diện tích hình tròn đó (giả sử người dùng đã nhập đúng).

R =

Diện tích: ...

4. Thoát khỏi chương trình

Sau khi thực hiện xong 1 yêu cầu của người dùng, chương trình phải hiển thị lại menu chọn. Chương trình chỉ chấm dứt khi người dùng chọn số 4.

Bài 02: Đọc số có hai chữ số. Tên chương trình: *Number.cpp*

Hãy viết chương trình nhập các số có hai chữ số từ màn hình, ghi chữ số đó dưới dạng chữ.

Dữ liệu vào từ màn hình:

- Người dùng nhập các số nguyên trong khoảng 10..99. Khi nào nhập số nằm ngoài khoảng này thì kết thúc việc nhập và đọc.

Kết quả ra màn hình:

- Tương ứng với các số nằm trong khoảng từ 10..99 in ra màn hình cách đọc tương ứng.

Dữ liệu vào	Kết quả tương ứng
number = 10	Mười
number = 15	Mười lăm
number = 20	Hai mươi
number = 99	Chín mươi chín
number = -1	

Bài 03: Xếp loại. Tên chương trình *Grade.cpp*

Viết chương trình nhập điểm thi từ bàn phím và hiển thị kết quả: kém nếu điểm 0, 1, 2 hoặc 3; Yếu nếu điểm là 4; Trung bình nếu điểm 5 hoặc 6; Khá nếu điểm 7 hoặc 8; Giỏi nếu điểm 9 hoặc 10.

Dữ liệu vào từ màn hình:

- Người dùng nhập các điểm số trong khoảng 0 ..10. Khi nào nhập số ngoài khoảng này thì kết thúc chương trình.

Kết quả ra màn hình:

- Tương ứng với điểm nhập vào thì in ra màn hình xếp loại của điểm số đó

Dữ liệu vào	Kết quả tương ứng
Diem = 0	Kem
Diem = 1	Kem
Diem = 2	Kem
Diem = 3	Kem
Diem = 4	Yeu
Diem = 5	Trung binh

Diem = 6	Trung bình
Diem = 7	Kha
Diem = 8	Kha
Diem = 9	Gioi
Diem = 10	Gioi

Mục đích:

- Sử dụng linh hoạt câu lệnh lặp với số lần lặp biết trước (for)
- Sử dụng các câu lệnh điều khiển vòng lặp với số lần lặp KHÔNG biết trước (while và do ...while)

Bài 01: In các số nguyên tố. Tên chương trình: *PrimeNumber.cpp*

Hãy in ra màn hình các số nguyên tố nhỏ hơn hoặc bằng n:

Dữ liệu vào từ màn hình:

- Nhập vào số nguyên dương n từ màn hình (giả thiết là người dùng nhập đúng).

Kết quả ra màn hình:

Dữ liệu vào	Kết quả tương ứng
n = 15	Cac so nguyen to nho hon hoac bang 15: 2 3 5 7 11 13

Bài 02: Tính e^x . Tên chương trình: **Taylor.cpp**

Cho khai triển Taylor của e^x như sau:

$$e^x = \sum_{k=0}^{\infty} \frac{x^k}{k!}$$

Ta có thể tính xấp xỉ e^x theo công thức sau:

$$e^x \approx 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots + \frac{x^n}{n!}$$

Hãy tính xấp xỉ e^x với sai số là: $\delta = \left| \frac{x^{n+1}}{(n+1)!} \right| \leq \text{err}$ (số err cho trước).

Dữ liệu vào từ màn hình:

- Nhập vào số thực x ($-1 \leq x \leq 1$)
- Nhập vào sai số err

Kết quả ra màn hình:

Dữ liệu vào	Kết quả tương ứng
$x = -0.7$ $\text{err} = 0.0001$	0.4966

Ghi chú:

Hàm mũ:

- `double pow (double base, double exponent);`
- `float pow (float base, float exponent);`
- `double pow (double base, int exponent);`

Hàm trị tuyệt đối của số thực:

- `double fabs (double x);`
- `float fabs (float x);`

Bài 03: Giải phương trình nghiệm nguyên. Tên chương trình: *Solve.cpp*

Tìm tất cả các nghiệm nguyên của phương trình: $a^2 + b^2 = c^2$. Với $m \leq a \leq b, c \leq n \leq 50$.

Dữ liệu vào từ màn hình:

- Nhập hai số nguyên dương m, n ($m \leq n \leq 50$).

Kết quả ra màn hình:

- In tất cả các bộ ba: a, b, c thỏa mãn phương trình trên.

Dữ liệu vào	Kết quả tương ứng
$m = 1$ $n = 20$	3 4 5 5 12 13 6 8 10 8 15 17 9 12 15

Bài 04: Tìm ước chung lớn nhất của 2 số. Tên chương trình là *Submul.cpp*

Tìm ước chung lớn nhất của hai số nguyên dương m và n (sử dụng thuật toán Euclid)

Thuật toán Euclid.

Input : m, n nguyên dương

Output : g , ước chung lớn nhất của m và n

Phương pháp :

Bước 1 : Tìm r , phần dư của phép chia m cho n

Bước 2 : Nếu $r = 0$, thì $g \leftarrow n$ (gán giá trị của n cho g) và dừng lại. Trong trường hợp ngược lại ($r \neq 0$), thì $m \leftarrow n, n \leftarrow r$ và quay lại bước 1.

Dữ liệu vào từ màn hình:

- Nhập hai số nguyên dương m và n (giả thiết người dùng nhập đúng) ($2 \leq m, n \leq 10^9$)

Kết quả ra màn hình:

- In ra ước chung lớn nhất của m và n .

Dữ liệu vào	Kết quả tương ứng
$m = 25$ $n = 15$	Ước chung lớn nhất của 25 và 15 là: 5

Bài 05: Kết hợp mệnh giá tiền. Tên chương trình **Money.cpp**

Có 5 loại giấy bạc 100, 200, 500, 1000. Viết chương trình hiển thị các phương án kết hợp các loại giấy bạc trên được số tiền N .

Dữ liệu vào:

- Nhập vào số nguyên dương N là bội số của 100 ($N \geq 100$)

Dữ liệu ra:

- In ra màn hình tất cả các phương án kết hợp có thể có, mỗi phương án trên một dòng

Dữ liệu vào	Dữ liệu ra tương ứng
$N=500$	Phương án 1: 0 to 100 + 0 to 200 + 1 to 500 + 0 to 1000 Phương án 2: 1 to 100 + 2 to 200 + 0 to 500 + 0 to 1000 Phương án 3: 3 to 100 + 1 to 200 + 0 to 500 + 0 to 1000 Phương án 4: 5 to 100 + 0 to 200 + 0 to 500 + 0 to 1000

Mở rộng:

- Giải quyết bài toán trong trường hợp N là một số nguyên dương bất kỳ (không nhất thiết là bội số của 100).

Bài 06: Giải hệ phương trình bậc nhất. Tên chương trình Slove.cpp

Hãy viết chương trình giải hệ phương trình bậc nhất:

$$\begin{cases} a_1x + b_1y + c_1 = 0 \\ a_2x + b_2y + c_2 = 0 \end{cases}$$

Dữ liệu vào từ màn hình:

- Đầu tiên nhập số k là số phương trình bậc nhất cần giải
- k dòng tiếp theo mỗi dòng ghi 6 số thực $a_1, b_1, c_1, a_2, b_2, c_2$. Các số cách nhau ít nhất một dấu cách trống.

Kết quả ra màn hình:

- Tương ứng với phương trình yêu cầu trong dữ liệu vào, ghi kết quả trong các trường hợp sau:
 - Phương trình vô nghiệm
 - Phương trình vô số nghiệm
 - Phương trình có nghiệm $x = a$ và $y = b$

Dữ liệu vào	Kết quả tương ứng
k = 3	Phương trình vô số nghiệm
1 1 1 2 2 2	Phương trình vô nghiệm
1 1 1 2 2 3	Phương trình có nghiệm $x = 1.000$ và $y = 1.000$
1 1 -2 2 2 -4	

Lưu ý: Các nghiệm ghi ra với độ chính xác 3 chữ số ở hàng thập phân.

Mục đích: Luyện tập các nội dung liên quan đến HÀM

- Định nghĩa chương trình con (Hàm) và sử dụng nó trong chương trình chính bằng lời gọi hàm (truyền giá trị). Chương trình con có một hoặc nhiều đối số.
- Định nghĩa hàm theo hai phương pháp: Không đệ quy và có đệ quy.

Bài 02: Tính e^x . Tên chương trình: **Taylor.cpp**

Cho khai triển Taylor của e^x như sau:

$$e^x = \sum_{k=0}^{\infty} \frac{x^k}{k!}$$

Ta có thể tính xấp xỉ e^x theo công thức sau:

$$e^x \approx 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots + \frac{x^n}{n!}$$

Hãy tính xấp xỉ e^x với sai số là: $\delta = \left| \frac{x^{n+1}}{(n+1)!} \right| \leq \text{err}$ (số err cho trước).

Dữ liệu vào từ màn hình:

- Nhập vào số thực x ($-1 \leq x \leq 1$)
- Nhập vào sai số err

Kết quả ra màn hình:

Dữ liệu vào	Kết quả tương ứng
$x = -0.7$ $\text{err} = 0.0001$	0.4966

Ghi chú:

Hàm mũ:

- `double pow (double base, double exponent);`
- `float pow (float base, float exponent);`
- `double pow (double base, int exponent);`

Hàm trị tuyệt đối của số thực:

- `double fabs (double x);`
- `float fabs (float x);`

Bài 03: Giải phương trình nghiệm nguyên. Tên chương trình: *Solve.cpp*

Tìm tất cả các nghiệm nguyên của phương trình: $a^2 + b^2 = c^2$. Với $m \leq a \leq b, c \leq n \leq 50$.

Dữ liệu vào từ màn hình:

- Nhập hai số nguyên dương m, n ($m \leq n \leq 50$).

Kết quả ra màn hình:

- In tất cả các bộ ba: a, b, c thỏa mãn phương trình trên.

Dữ liệu vào	Kết quả tương ứng
m = 1 n = 20	3 4 5 5 12 13 6 8 10 8 15 17 9 12 15

Bài 04: Tìm ước chung lớn nhất của 2 số. Tên chương trình là *Submul.cpp*

Tìm ước chung lớn nhất của hai số nguyên dương m và n (sử dụng thuật toán Euclid)

Thuật toán Euclid.

Input : m, n nguyên dương

Output : g, ước chung lớn nhất của m và n

Phương pháp :

Bước 1 : Tìm r, phần dư của phép chia m cho n

Bước 2 : Nếu $r = 0$, thì $g \leftarrow n$ (gán giá trị của n cho g) và dừng lại. Trong trường hợp ngược lại ($r \neq 0$), thì $m \leftarrow n, n \leftarrow r$ và quay lại bước 1.

Dữ liệu vào từ màn hình:

- Nhập hai số nguyên dương m và n (giả thiết người dùng nhập đúng) ($2 \leq m, n \leq 10^9$)

Kết quả ra màn hình:

- In ra ước chung lớn nhất của m và n .

Dữ liệu vào	Kết quả tương ứng
$m = 25$ $n = 15$	Ước chung lớn nhất của 25 và 15 là: 5

Bài 05: Kết hợp mệnh giá tiền. Tên chương trình **Money.cpp**

Có 5 loại giấy bạc 100, 200, 500, 1000. Viết chương trình hiển thị các phương án kết hợp các loại giấy bạc trên được số tiền N .

Dữ liệu vào:

- Nhập vào số nguyên dương N là bội số của 100 ($N \geq 100$)

Dữ liệu ra:

- In ra màn hình tất cả các phương án kết hợp có thể có, mỗi phương án trên một dòng

Dữ liệu vào	Dữ liệu ra tương ứng
$N=500$	Phương án 1: 0 to 100 + 0 to 200 + 1 to 500 + 0 to 1000 Phương án 2: 1 to 100 + 2 to 200 + 0 to 500 + 0 to 1000 Phương án 3: 3 to 100 + 1 to 200 + 0 to 500 + 0 to 1000 Phương án 4: 5 to 100 + 0 to 200 + 0 to 500 + 0 to 1000

Mở rộng:

- Giải quyết bài toán trong trường hợp N là một số nguyên dương bất kỳ (không nhất thiết là bội số của 100).

Bài 06: Giải hệ phương trình bậc nhất. Tên chương trình Slove.cpp

Hãy viết chương trình giải hệ phương trình bậc nhất:

$$\begin{cases} a_1x + b_1y + c_1 = 0 \\ a_2x + b_2y + c_2 = 0 \end{cases}$$

Dữ liệu vào từ màn hình:

- Đầu tiên nhập số k là số phương trình bậc nhất cần giải
- k dòng tiếp theo mỗi dòng ghi 6 số thực $a_1, b_1, c_1, a_2, b_2, c_2$. Các số cách nhau ít nhất một dấu cách trống.

Kết quả ra màn hình:

- Tương ứng với phương trình yêu cầu trong dữ liệu vào, ghi kết quả trong các trường hợp sau:
 - Phương trình vô nghiệm
 - Phương trình vô số nghiệm
 - Phương trình có nghiệm $x = a$ và $y = b$

Dữ liệu vào	Kết quả tương ứng
k = 3	Phương trình vô số nghiệm
1 1 1 2 2 2	Phương trình vô nghiệm
1 1 1 2 2 3	Phương trình có nghiệm $x = 1.000$ và $y = 1.000$
1 1 -2 2 2 -4	

Lưu ý: Các nghiệm ghi ra với độ chính xác 3 chữ số ở hàng thập phân.

BÀI THỰC HÀNH SỐ 4:: CẤU TRÚC LỆNH

A. CÂU HỎI

1. Cho biết số các bước lặp nhỏ nhất của câu lệnh while và lệnh do...while?
2. Giả sử s là một câu lệnh có cấu trúc và e1, e2, e3 là các biểu thức, có sự khác nhau gì giữa đoạn mã:

```
for(e1;e2;e3)    s;
```

và đoạn mã:

```
e1;
while(e2) { s;
           e3;
        }
```

3. Cho biết giá trị của biến x sau khi thực hiện câu lệnh sau:

```
for(x = 0; x < 100; x++);
```

4. Cho biết giá trị của biến ctr sau khi thực hiện câu lệnh sau:

```
for(ctr = 2; ctr < 10; ctr += 3);
```

5. Có bao nhiêu ký tự X được in ra sau khi thực hiện đoạn mã:

```
for(x = 0; x < 10; x++)
for(y = 5; y > 0; y--)puts("X");
```

6. Cho biết điểm sai trong đoạn mã sau:

```
record = 0;
while(record < 100)
{ printf("\nRecord %d ", record);
  printf("\nGetting next number...");
}
```

7. Có nhận xét gì về đoạn mã sau:

```
for(counter = 1; counter < 10; counter++);
printf("\nCounter = %d", counter);
```

8. Sự khác nhau giữa lệnh break và lệnh continue là gì?
9. Vòng lặp vô hạn là gì? Làm thế nào để tạo nó?
10. Hai sự kiện gì làm kết thúc việc chạy chương trình?
11. Hàm exit() dùng để làm gì?
12. Câu lệnh gì làm cho điều khiển chương trình chuyển đến bước lặp kế tiếp trong một lệnh lặp?
13. Câu lệnh gì chuyển điều khiển chương trình đến cuối vòng lặp?
14. Có nhận xét gì về đoạn mã sau?

```
switch(answer)
{ case 'Y': printf("You answered yes");
  break;
  case 'N': printf("You answered no");
}
```

15. Có nhận xét gì về đoạn mã sau?

```
switch(choice)
{default:
    printf("You did not choose 1 or 2");
case 1:
    printf("You answered 1");
    break;
case 2:
    printf("You answered 2");
    break;
}
```

B. Bài tập:

1. Viết chương trình nhập dữ liệu cho các biến kiểu int, long int, float, chuỗi, sau đó hiển thị ra màn hình.

2. Viết chương trình minh họa việc khai báo và khởi gán các biến.

3. Cho biết kết quả của chương trình sau đây:

```
#include<stdio.h>
#include<conio.h>
void main()
{int m = 3,p = 5;  int a1,a2,a3,a4,a5;  float x1,x2,x3,x4;
printf("\n Tim gia tri gan cho cac bien ");
a1 = m<p;
a2 = m == p;
a3 = p%m + p>m;
a4 = m*(p>m ? m:p);
a5 = m*(p<m ? p:p);
x1 = p/m;
x2 = (float)p/m;
x3 = (p +0.5)/m;
x4 = (int)(p+0.5)/m;
printf("\n a1 = %d ",a1);
printf("\n a2 = %d ",a2);
printf("\n a3 = %d ",a3);
printf("\n a4 = %d ",a4);
printf("\n a5 = %d ",a5);
printf("\n x1 = %10.3f ",x1);
printf("\n x2 = %10.3f ",x2);
printf("\n x3 = %10.3f ",x3);
printf("\n x4 = %10.3f ",x4);
getch();
}
```

}

4. Cho biết kết quả của chương trình sau đây:

```
#include <stdio.h>
#include <conio.h>
void main()
{int a = 10, b = 5, c = 10, d;
 clrscr();
 printf("\n Minh hoa phep toan tang giam \n");
 d=a==(b=c);
 printf("A:a =%d b =%d c =%d d =%d\n",a,b,c,d);
 a=b=c=5;
 a+=b+=c;
 printf("B:a =%d b =%d c =%d \n",a,b,c);
 c=a<b?a++:b++;
 printf("C:a =%d b =%d c =%d \n",a,b,c);
 c=a>b?a++:b++;
 printf("D:a =%d b =%d c =%d \n",a,b,c);
 getch();
}
```

5. Cho biết kết quả của chương trình sau đây:

```
#include<stdio.h>
#include<conio.h>
void main()
{int a,b,c;
 clrscr();
 printf("\n Chuong trinh minh hoa toan tu logic \n ");
 a = 5; b = 2; /* Truong hop 1 */
 c = (a++ >b) || (b++ != 3);
 printf("A: a = %d b = %d c = %d\n",a,b,c);
 a = 5; b = 2 ; /* Truong hop 2 */
 printf("B: a = %d b = %d c = %d\n",a,b,c);
 a = 5; b = 2 ; /* Truong hop 3 */
 c = (++a == 3) && (++b == 3);
 printf("C: a = %d b = %d c = %d\n",a,b,c);
 a = 5; b = 2; /* Truong hop 4 */
 c = (++a == 6) && (++b == 3);
 printf("D: a = %d b = %d c = %d\n",a,b,c);
 getch();
}
```

6. Viết chương trình tìm số lớn nhất và nhỏ nhất trong 4 số thực nhập từ bàn phím.

7. Viết chương trình tìm x^y

- 8.** Viết chương trình tính $c = a / b$, với a và b là hai số nguyên, c số thực.
- 9.** Viết chương trình tìm $c = a/b$, với a, b, c nguyên, hiển thị ở các số hệ 8, 16 và 10.
- 10.** Viết chương trình nhập điểm thi từ bàn phím và hiển thị kết quả: kém nếu điểm từ 0 đến cận 3; Yếu nếu điểm từ 3 đến cận 5; Trung bình nếu điểm từ 5 đến cận 7; Khá nếu điểm từ 7 đến cận 9; Giỏi nếu điểm từ 9 đến 10.
- 11.** Viết chương trình nhập điểm thi từ bàn phím và hiển thị kết quả: kém nếu điểm 0, 1, 2 hoặc 3; Yếu nếu điểm là 4; Trung bình nếu điểm 5 hoặc 6; Khá nếu điểm 7 hoặc 8; Giỏi nếu điểm 9 hoặc 10.
- 14.** Có 5 loại giấy bạc 200, 500, 1000, 2000, 5000, 10000, 20000. Viết chương trình hiển thị các phương án kết hợp các loại giấy bạc trên cho ra 100000.
- 15.** Viết chương trình tính tổng nghịch đảo của n số tự nhiên đầu tiên.
- 16.** Viết chương trình tìm và in ra các số không vượt quá 1000 và thỏa tính chất: số bằng tổng các ước số của nó mà không kể chính nó. Ví dụ: $6 = 1 + 2 + 3$.
- 17.** Viết chương trình tìm các ước số nguyên tố của số nguyên dương N cho trước.
- 18.** Viết chương trình tính căn bậc 2 của số nguyên a không âm cho trước theo công thức lặp sau:
$$x_0 = a, \quad x_{n+1} = (x_n + a/x_n)/2$$
Quá trình lặp sẽ dừng khi $|x_{n+1} - x_n| < \varepsilon$. Với ε là số đủ bé cho trước.

BÀI THỰC HÀNH SỐ 5:: HÀM

A. CÂU HỎI

1. Câu hỏi về hàm:

1. Dòng đầu tiên của định nghĩa hàm gọi là gì, nó bao gồm các thông tin thế nào?
2. Hàm có thể trả về bao nhiêu giá trị?
3. Nếu một hàm không trả về giá trị, kiểu gì có thể dùng để khai báo hàm?
4. Sự khác nhau giữa định nghĩa hàm và nguyên mẫu hàm?
5. Biến địa phương là gì? Phạm vi của biến địa phương?
6. Khi nào nên sử dụng biến toàn cục?
7. Viết một tiêu đề hàm có tên là `do_it()` có 3 đối số kiểu char và trả về kiểu float.
8. Viết một tiêu đề hàm có tên là `print_a_number()` có 1 đối số kiểu int và không trả về điều gì.
9. Tìm điểm sai trong đoạn mã sau và sửa lại cho đúng:

```
#include <stdio.h>
void print_msg(void);
main()
{ print_msg("This is a message to print");
  return 0;
}
void print_msg(void)
{ puts("This is a message to print");
  return 0;
}
```

10. Tìm điểm sai trong định nghĩa hàm sau đây:

```
int twice(int y);
{return (2 * y);}
```

11. Xem chương trình sau đây:

```
#include <stdio.h>
int x, y, z;
int larger_of(int , int);
main()
{puts("Enter two different integer values: ");
 scanf("%d%d", &x, &y);
 z = larger_of(x,y);
 printf("\nThe larger value is %d.", z);
 return 0;
}
int larger_of(int a, int b)
{if(a > b) return a;
 else return b;
}
```

Hãy viết lại hàm `larger_of()` trong chương trình này mà chỉ dùng một câu lệnh `return` duy nhất.

2. Câu hỏi về lớp lưu trữ biến

1. Sự khác nhau quan trọng nhất giữa lớp lưu trữ địa phương và lớp lưu trữ ngoài là gì?
2. Vị trí của biến ảnh hưởng đến lớp lưu trữ của nó như thế nào?
3. Khi định nghĩa một biến địa phương, hai tùy chọn gì ảnh hưởng đối với thời gian tồn tại của biến?
4. Chương trình có thể khởi tạo các biến tĩnh và tự động khi chúng được định nghĩa. Việc khởi tạo thực hiện khi nào?
5. Biến thanh ghi luôn luôn được đặt trong thanh ghi. Đúng hay sai?
6. Biến toàn cục không được khởi tạo chứa giá trị gì?
7. Biến địa phương không được khởi tạo chứa giá trị gì?
8. Một hàm cần ghi nhận giá trị của biến địa phương kiểu `int` giữa các lần gọi thì biến nên được khai báo thế nào?
9. Viết một khai báo cho biến thanh ghi kiểu `int`.
10. Sửa lại chương trình sau cho khỏi bị lỗi:

```
#include <stdio.h>
void print_value(void);
main()
{
    int x = 999;
    printf("%d\n", x);
    print_value();
    return 0;
}
void print_value(void)
{
    printf("%d\n", x);
}
```

11. Viết chương trình khai báo một biến toàn cục kiểu `int` có tên là `var`. Khởi tạo cho biến `var` một giá trị nào đó. Chương trình in giá trị của `var` trong một hàm nào đó (không phải hàm `main`). Có cần phải truyền `var` như là tham số của hàm hay không?
12. Sửa lại chương trình ở ví dụ trên, trong đó `var` là biến địa phương trong hàm `main()`. Chương trình vẫn in giá trị của `var` trong một hàm nào đó (không phải hàm `main()`). Có cần phải truyền `var` như là tham số của hàm hay không?
13. Trong chương trình thì biến toàn cục và biến địa phương trùng tên được không? Viết chương trình minh họa.
14. Có điểm gì sai trong đoạn mã sau:

```
void a_sample_function(void)
{
    int ctrl;
    for(ctrl = 0; ctrl < 25; ctrl++)
        printf("*");
    puts("\nThis is a sample function");
    {
        char star = '*';
        puts("\nIt has a problem\n");
        for(int ctr2 = 0; ctr2 < 25; ctr2++)
```

```
    {  
        printf("%c", star);  
    }  
}
```

15. Có điểm gì sai trong đoạn mã sau:

```
#include <stdio.h>  
main()  
{int x = 1;  
  static int tally = 0;  
  for(x = 0; x < 101; x++)  
      {if(x % 2 == 0)tally++;  
      }  
  printf("There are %d even numbers.\n", tally);  
  return 0;  
}
```

16. Có điểm gì sai trong đoạn mã sau:

```
#include <stdio.h>  
void print_function(char star);  
int ctr;  
main()  
{char star;  
  print_function(star);  
  return 0;  
}  
void print_function(char star)  
{char dash;  
  for(ctr = 0; ctr < 25; ctr++)  
      {printf("%c%c", star, dash);}  
}
```

17. Chương trình sau cho kết quả gì?

```
#include <stdio.h>  
void print_letter2(void);  
int ctr;  
char letter1 = 'X';  
char letter2 = '=';  
main()  
{for(ctr = 0; ctr < 10; ctr++)  
    { printf("%c", letter1);  
      print_letter2();}  
  return 0;
```

```

    }
void print_letter2(void)
{for(ctr = 0; ctr < 2; ctr++)
    printf("%c", letter2);
}

```

18. Viết lại chương trình trên.

B. BÀI TẬP

- Viết hàm tìm số lớn nhất trong 3 số thực.
- Viết hàm kiểm tra năm nhuận.
- Viết hàm kiểm tra số nguyên tố.
- Viết hàm đệ quy và hàm không đệ quy để tìm số hạng thứ N của dãy số Fibonacci được định nghĩa như sau:

$$f_0 = f_1 = 1, \quad f_{n+1} = f_n + f_{n-1} \quad (n = 1, 2, \dots)$$

- Viết hàm đệ quy và không đệ quy để tính giai thừa của số nguyên n không âm.
- Viết hàm để giải phương trình bậc hai: $ax^2 + bx + c = 0$ (a khác 0)
- Viết hàm để tính $\cos x$ theo công thức sau:

$$\cos x \approx 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \dots + (-1)^n \frac{x^{2n}}{(2n)!} + \dots$$

Quá trình lặp sẽ dừng khi $\left| (-1)^n \frac{x^{2n}}{(2n)!} \right| < \varepsilon$ với ε là số đủ bé cho trước.

- Viết hàm để tính $\sin x$ theo công thức sau:

$$\sin x \approx x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots + (-1)^n \frac{x^{2n+1}}{(2n+1)!} + \dots$$

Quá trình lặp sẽ dừng khi $\left| (-1)^n \frac{x^{2n+1}}{(2n+1)!} \right| < \varepsilon$ với ε là số đủ bé cho trước.

- Viết hàm để tính e^x theo công thức sau:

$$e^x \approx 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots + \frac{x^n}{n!} + \dots$$

Quá trình lặp sẽ dừng khi $\left| \frac{x^n}{n!} \right| < \varepsilon$ với ε là số đủ bé cho trước.

- Viết hàm để tính số π theo công thức sau:

$$\frac{\pi}{4} \approx 1 - \frac{1}{3} + \frac{1}{5} - \dots + (-1)^n \frac{1}{2n+1} + \dots$$

Quá trình lặp sẽ dừng khi $\left| (-1)^n \frac{1}{2n+1} \right| < \varepsilon$ với ε là số đủ bé cho trước.

- Viết hàm để tính căn bậc hai của một số không âm a theo công thức lặp:

$$x_0 = a, \quad x_{n+1} = (x_n + a/x_n)/2$$

Quá trình lặp sẽ dừng khi $|x_{n+1} - x_n| < \varepsilon$ với ε là số đủ bé cho trước. Lúc đó:

$$\sqrt{a} \approx x_n$$

12. Viết hàm để liệt kê các số nguyên tố không vượt quá số nguyên dương N cho trước.

13. Viết hàm đệ quy để tính x^n theo hai cách khác nhau.

14. Viết hàm đệ quy để tìm ước số chung lớn nhất của hai số nguyên dương theo hai cách khác nhau.

15. Viết hàm để in tam giác Pascal ra màn hình.

16. Viết hàm để tính $S = \sqrt{k + \underbrace{\sqrt{k + \sqrt{k + \dots + \sqrt{k}}}}_{n \text{ dấu căn}}}$

17. Viết chương trình để giải bài toán tháp Hà Nội: Có n đĩa được sắp xếp trên một cọc A có kích thước nhỏ dần (lớn dưới nhỏ trên). Yêu cầu đặt ra là: Chuyển chồng đĩa từ cọc A sang cọc C theo những điều kiện:

- Mỗi lần chỉ chuyển một đĩa.
- Không có tình huống đĩa lớn trên đĩa nhỏ (dù chỉ là tạm thời).
- Được phép sử dụng một cọc B làm cọc trung gian để đặt tạm đĩa khi chuyển từ cọc A sang cọc C.

BÀI THỰC HÀNH SỐ 6 : MẢNG, CHUỖI VÀ CON TRỎ

A. CÂU HỎI**Câu hỏi về mảng:**

1. Các kiểu dữ liệu nào của C có thể được dùng trong mảng?
2. Điều gì xảy ra nếu chương trình truy cập đến một phần tử có chỉ số nằm ngoài phạm vi mảng?
3. Mảng được khai báo sau đây có bao nhiêu phần tử?

```
int array[2][3][5][8];
```

Tên của phần tử thứ 10 là gì?

4. Viết lệnh khai báo một mảng nguyên có 10 phần tử và khởi tạo tất cả các phần tử là 1.
5. Cho mảng sau, viết mã để khởi tạo tất cả các phần tử là 88:

```
int eightyeight[88];
```

6. Cho mảng sau, viết mã để khởi tạo tất cả các phần tử là 0:

```
int stuff[12][10];
```

7. Tìm điểm sai trong đoạn mã sau:

```
int x, y;
int array[10][3];
main()
{
    for(x = 0; x < 3; x++)
        for(y = 0; y < 10; y++)
            array[x][y] = 0;
    return 0;
}
```

8. Tìm điểm sai trong đoạn mã sau:

```
int array[10];
int x = 1;
main()
{
    for(x = 1; x <= 10; x++)
        array[x] = 99;
    return 0;
}
```

9. Viết chương trình tạo ra các số ngẫu nhiên cho mảng a[5][4]. Hiển thị ra màn hình các giá trị của mảng theo từng cột.
10. Viết chương trình khởi tạo mảng một chiều gồm 1000 số ngẫu nhiên. Hiển thị giá trị trung bình và giá trị các phần tử mảng, và sau mỗi 10 giá trị hiển thị thì tạm dừng màn hình.
11. Viết chương trình khởi tạo mảng một chiều gồm 10 phần tử, mỗi phần tử có giá trị bằng chỉ số của nó.
12. Sửa lại chương trình ở câu 9, sau khi hiển thị các giá trị khởi tạo, chương trình sẽ sao chép các giá trị vào một mảng mới và thêm vào 10 cho mỗi giá trị. Hiển thị ra màn hình giá trị các phần tử của mảng mới.

Câu hỏi về con trỏ:

1. Toán tử gì được dùng để xác định địa chỉ của một biến?
2. Toán tử gì được dùng để xác định giá trị ở vị trí được trỏ bởi một con trỏ?
3. Con trỏ là gì?
4. Truy cập gián tiếp là gì?
5. Mảng được lưu trữ trong bộ nhớ như thế nào?
6. Chỉ ra hai cách để nhận được địa chỉ phần tử đầu tiên của mảng data[].
7. Nếu mảng được truyền đến một hàm, hai cách gì để nhận biết mảng kết thúc ở đâu?
8. Sáu toán tử gì có thể thực hiện với con trỏ?
9. Giả sử bạn có hai con trỏ. Nếu con trỏ đầu tiên trỏ đến phần tử thứ ba trong một mảng kiểu int, con trỏ thứ hai trỏ đến phần tử thứ tư. Việc trừ con trỏ thứ hai cho con trỏ đầu cho kết quả gì?
10. Giả sử cost là một tên biến.

Làm thế nào để khai báo và khởi tạo một con trỏ có tên p_cost trỏ đến biến đó.

Làm thế nào để gán giá trị 100 cho biến cost bằng cách dùng cả truy cập trực tiếp và gián tiếp.

Làm thế nào để in giá trị của con trỏ p_cost và giá trị con trỏ p_cost trỏ đến.

11. Làm thế nào để gán địa chỉ của biến thực có tên là radius cho một biến con trỏ.
12. Hai cách để gán giá trị 100 cho phần tử thứ ba của mảng data[].
13. Viết một hàm có tên là sumarrays() có đối số là hai mảng, tính tổng giá trị cả hai mảng và trả về tổng đó. Viết chương trình minh họa.
14. Viết lệnh khai báo biến con trỏ, khai báo và khởi gán con trỏ trỏ tới biến, khai báo và khởi gán con trỏ trỏ đến con trỏ.

15. Xét khai báo

```
float x;  
float *px = &x;  
float **ppx = &px;
```

Để gán 100 cho biến x, ta có thể viết như sau hay không?

```
*ppx = 100;
```

16. Viết một nguyên mẫu hàm với đối là một mảng con trỏ kiểu char và trả về void.
17. Con trỏ trỏ đến hàm là gì?
18. Viết một khai báo con trỏ trỏ đến hàm trả về kiểu char và có đối là một mảng con trỏ kiểu char.
19. Khai báo sau có gì sai:

```
char *ptr(char *x[]);
```

20. Giải thích các khai báo sau:

- a. int *var1;
- b. int var2;
- c. int **var3;

21. Giải thích các khai báo sau:

- a. int a[3][12];
- b. int (*b)[12];
- c. int *c[12];

22. Giải thích các khai báo sau:

- a. `char *z[10];`
- b. `char *y(int field);`
- c. `char (*x)(int field);`

23. Viết một khai báo con trỏ trỏ đến hàm có đối kiểu nguyên và trả về biến kiểu float.

24 Viết một khai báo mảng con trỏ trỏ đến hàm có đối là chuỗi ký tự và trả về số nguyên.

25. Viết lệnh khai báo mảng 10 con trỏ kiểu char.

26. Có điểm gì sai trong đoạn mã sau:

```
int x[3][12];
int *ptr[12];
ptr = x;
```

B. BÀI TẬP

1. Khởi tạo mảng các số ngẫu nhiên từ 0 đến 127.

2. Cho biết kết quả của việc thực hiện chương trình sau:

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int ar[]={10,15,4,25,3,-4},*p, i;
    clrscr();
    p=&ar[2];
    printf("Cac gia tri cua mang:\n\n");
    for(i=0;i<6;i++)printf("ar[%d]=%4d\n",i,ar[i]);
    printf("\nCho biet gia tri cua cac bieu thuc\n\n");
    printf("a) *(p+1)= %d\n\n",*(p+1));
    printf("b) p[-1]= %d\n\n",p[-1]);
    printf("c) (ar-p)=%d\n\n", (ar-p));
    printf("d) ar[*p++]=%d\n\n",ar[*p++]);
    printf("e) *(ar+ar[2])=%d\n\n",*(ar+ar[2]));
    getch();
}
```

3. Viết chương trình khởi tạo mảng ngẫu nhiên các số nguyên có kích thước không quá 100 và sắp xếp theo thứ tự tăng.

4. Viết chương trình nhập một mảng n số thực bằng cách dùng biến con trỏ, cấp phát bộ nhớ cho mảng rồi sắp xếp theo thứ tự tăng dần.

5. Viết chương trình nhập một mảng thực cấp không quá 20x20 và tìm giá trị lớn nhất trong các phần tử của mảng.

6. Hãy viết một hàm gọi là *merge_arrays()* nhận vào hai mảng một chiều đã được sắp xếp và trộn chúng thành một mảng cũng được sắp xếp. Khi thực hiện, không được chuyển hai mảng vào một mảng rồi sắp xếp lại. Hàm có tiêu đề như sau:

```
void merge_arrays(double a[],double b[],double c[], int n, int m)
```

ở đây, a và b là hai mảng đã sắp xếp và c là mảng chứa kết quả trộn.

7. Hãy viết chương trình thực hiện các công việc sau:

- Khởi tạo mảng ngẫu nhiên.
- Thông báo lên màn hình phần tử bé nhất và lớn nhất trong dãy.
- Thông báo lên màn hình tổng các số âm và trung bình cộng các số dương của dãy trên
- Nhập vào một số nguyên x . Thông báo lên màn hình số nguyên x đó có trong dãy trên hay không.
- Kiểm tra xem dãy ở trên đã được sắp xếp tăng dần chưa, nếu chưa thì tiến hành sắp xếp dãy nó tăng dần và in dãy sau khi sắp xếp ra màn hình.

8. Cho một mảng A gồm n phần tử nguyên đã được sắp xếp (tăng dần hoặc giảm dần). Nhập vào một số nguyên x , tiến hành chèn số x này vào mảng A sao cho thứ tự trong mảng không thay đổi.

9. Cho một ma trận A cấp $m \times n$ của các số nguyên. Viết chương trình thực hiện các công việc sau:

- Thông báo lên màn hình phần tử lớn nhất và bé nhất của ma trận A .
- Đếm xem trong ma trận A có bao nhiêu phần tử bằng với số nguyên x được nhập từ bàn phím và thông báo vị trí của chúng
- Tiến hành sắp xếp lại ma trận A theo thứ tự tăng dần của mỗi hàng và in ra màn hình ma trận A sau khi sắp xếp.

10. Viết chương trình in tất cả phương án chia n viên bi cho m người.

11. Viết chương trình liệt kê tất cả các hoán vị của tập $\{1, 2, \dots, n\}$

12. Nhập dãy số thực từ bàn phím, sắp xếp dãy theo thứ tự tăng dần và hiển thị ra màn hình.

13. Viết chương trình sau nhập một mảng thực kích thước không quá 20×20 , in mảng đã nhập và các phần tử lớn nhất và nhỏ nhất trên mỗi hàng của mảng.

14. Viết chương trình tính tổng theo hàng của ma trận cấp 3×5 .

15. Cho biết kết quả việc thực hiện chương trình sau:

```
#include <iostream.h>
#include <conio.h>
void main()
{
    clrscr();
    int x=10;
    int *px;
    px=&x; //con tro px tro toi bien x
    cout<<"\ndia chi cua bien x la:"<<&x;
    cout<<"\ndia chi cua con tro px la:"<<&px;
    cout<<"\noi dung cua bien x la:"<<x;
    cout<<"\noi dung cua px (chua dia chi bien x) la:"<<px;
    cout<<"\nGia tri ma px tro den:"<<*px;
    getch();
}
```

16. Cho biết kết quả việc thực hiện chương trình sau:

```
#include <iostream.h>
#include <conio.h>
void main()
{
```

```

clrscr();
int k,a[10],*p,*q;
for(k=0;k<10;k++)a[k]=3*k+5;
p=a;
q=p;
for(k=0;k<3;k++)cout<<*(q+k)<<' ';
p=q+2;
cout<<"\n";
for(k=0;k<3;k++)cout<<*(p+k)<<' ';
getch();
}

```

17. Giải thích cách thức hoạt động của con trỏ hàm trong chương trình sau đây:

```

#include <stdio.h>
#include <conio.h>
#include <iostream.h>
enum bool {false, true};
void nhap(int &x, int &y);
void bp(int &x, int &y);
void lp(int &x, int &y);
void hv(int &x, int &y);
void hienthi(int a, int b);
void main()
{ void (*pf)(int &, int &);
  bool thoat = false;  int a=2, b=4;  int chon;
  clrscr();
  while(thoat == false)
  {printf("\n (0)Thoat    (1)Nhap    (2)Binh phuong    (3)Lap
    phuong    (4)Hoan vi \n"); scanf("%d",&chon);
    switch (chon)
    {case 1: pf = nhap;break;
     case 2: pf = bp;break;
     case 3: pf = lp;break;
     case 4: pf = hv;break;
     default: thoat = true;break;
    }
    if(thoat)break;
    hienthi(a,b);
    pf(a,b);
    hienthi(a,b);
  }
  getch();
}

```

```
    }
void hienthi(int x, int y)
    { printf("\n a = %d, b = %d \n",x,y); }
void bp(int &x, int &y)
    {
        x *= x;
        y *= y;
    }
void lp(int &x, int &y)
    {
        x = x*x*x;
        y = y*y*y;
    }
void hv(int &x, int &y)
    { int tam =x;
        x = y;
        y = tam;
    }
void nhap(int &x, int &y)
    { printf("\n Nhap gia tri moi cua a: "); scanf("%d",&x);
        printf("\n Nhap gia tri moi cua b: "); scanf("%d",&y);
    }
```

18. Viết lại chương trình trên nhưng không sử dụng con trỏ hàm.

19. Giải thích cách thức hoạt động của mảng con trỏ hàm trong chương trình sau đây:

```
#include <stdio.h>
#include <conio.h>
#include <iostream.h>
void nhap(long &x, long &y);
void bp(long &x, long &y);
void lp(long &x, long &y);
void hv(long &x, long &y);
void hienthi(long a, long b);
void main()
{
    const max = 5;
    void (*pfa[max])(long &, long &);
    long a=1, b=2;
    int chon;
    clrscr();
    for(int i=0;i<max;i++)
        {printf("\n(1)Nhap (2)Binhphuong (3)Lapphuong (4)Hoanvi \n");
```

```
scanf("%d",&chon);
switch (chon)
{
    case 1: pfa[i] = nhap;break;
    case 2: pfa[i] = bp;break;
    case 3: pfa[i] = lp;break;
    case 4: pfa[i] = hv;break;
    default:pfa[i] = 0;
}
}
for(i=0;i<max;i++)
{
    pfa[i](a,b);
    hienthi(a,b);
}
getch();
}
void hienthi(long x, long y)
{printf("\n a = %ld, b = %ld \n",x,y);}
void bp(long &x, long &y)
{
    x *= x;
    y *= y;
}
void lp(long &x, long &y)
{
    x = x*x*x;
    y = y*y*y;
}
void hv(long &x, long &y)
{ long tam =x;
  x = y;
  y = tam;
}
void nhap(long &x, long &y)
{ printf("\n Nhập gia tri moi cua a: ");
  scanf("%ld",&x);
  printf("\n Nhập gia tri moi cua b: ");
  scanf("%ld",&y);
}
```

C. CÂU HỎI VÀ BÀI TẬP VỀ CHUỖI

1. Viết chương trình bỏ ký tự **a** trong một chuỗi.
2. Viết hàm trích chuỗi con bên trái của một chuỗi.
3. Viết hàm trích chuỗi con bên phải của một chuỗi.
4. Viết hàm trích chuỗi con trong một chuỗi.
5. Viết hàm xóa chuỗi con trong một chuỗi.
6. Viết hàm in các từ của chuỗi trên mỗi dòng.
7. Viết hàm tìm vị trí một chuỗi con trong một chuỗi.
8. Viết hàm ghép hai chuỗi thành một chuỗi.
9. Viết chương trình nhập vào tên của không quá 50 học sinh, sắp xếp và in ra theo thứ tự tăng. Chương trình sẽ dùng một hàm nhập dữ liệu, một hàm so sánh chuỗi và một hàm hiển thị kết quả sắp xếp.
10. Viết hàm đảo ngược một chuỗi.
11. Viết hàm để kiểm tra chuỗi có đối xứng hay không?
12. Viết hàm để loại bỏ các khoảng trống thừa trong một chuỗi.
13. Nhập một chuỗi từ bàn phím, kết thúc khi bấm Ctrl-Z và Enter. Đếm số từ trong các dòng đã nhập. Dòng tiêu đề của hàm đếm số từ như sau:

```
int NumberWords(char szString[])
```
14. Cùng câu hỏi bài 13 nhưng dòng tiêu đề đếm số từ như sau:

```
int NumberWords(char * pString)
```
15. Viết chương trình in các dòng được nhập vào từ bàn phím, kết thúc bằng cách bấm Ctrl-Z và Enter.
16. Viết hàm để tìm và thay thế một chuỗi con trong chuỗi cho trước.
17. Viết chương trình tìm số dòng, số từ, số ký tự của một chuỗi. Chuỗi được nhập vào từ bàn phím và kết thúc bằng cách bấm Ctrl-Z.
18. Viết chương trình tìm những người có họ là “Nguyen” trong một danh sách họ tên cho trước.
19. Viết hàm tìm từ dài nhất trong một chuỗi.

BÀI THỰC HÀNH SỐ 7:: KIỂU CẤU TRÚC**A. CÂU HỎI**

1. Sự khác nhau giữa cấu trúc và mảng?

2. Đoạn mã sau làm việc gì?

```
struct address {
    char name[31];
    char add1[31];
    char add2[31];
    char city[11];
    char state[3];
    char zip[11];
}myaddress = {"Bradley Jones", "RTSoftware", "P.O. Box 1213",
"Carmel", "IN", "46032-1213"};
```

3. Giả sử bạn khai báo một mảng cấu trúc và ptr là một con trỏ trỏ đến phần tử đầu tiên của mảng. Làm thế nào để ptr trỏ sang phần tử thứ hai của mảng.

4. Có điểm gì sai trong đoạn mã sau?

```
struct {
    char zodiac_sign[21];
    int month;
} sign = "Leo", 8;
```

5. Có điểm gì sai trong đoạn mã sau?

```
/* setting up a union */
union data{
    char a_word[4];
    long a_number;
}generic_variable = { "WOW", 1000 };
```

B. BÀI TẬP

1. Viết chương trình thực hiện các toán tử trên phân số: cộng, trừ, nhân, chia phân số.

2. Để quản lý điểm thi cho một lớp học Tin học Văn phòng, người ta cần quản lý các thông tin sau về mỗi học viên:

```
HoLot          Chuỗi ký tự (30)    /* Họ lót */
Ten            Chuỗi ký tự (30)    /* Tên */
Ngaysinh       Chuỗi ký tự (8)     /* Ngày sinh */
Noisinh        Chuỗi ký tự (40)    /* Nơi sinh */
D_WIN          0..10                /*ĐiểmWindows,điểm tròn
*/
D_WORD         0..10                /* Điểm Word, điểm tròn
*/
D_EXCEL        0..10                /* Điểm Excel, điểm tròn
*/
```

TONGDIEM 0..30 /* Tổng điểm */

X_LOAI Chuỗi ký tự (20) /* Xếp loại */

Hãy lập chương trình thực hiện các công việc sau:

- Nhập thông tin của học viên khi đến ghi danh. (Chỉ nhập *HoLot*, *Ten* và *Ngaysinh*).
- Nhập điểm các môn sau khi thi xong: Có thể nhập điểm cho bất kỳ môn nào tại mỗi thời điểm.
- In danh sách học viên ra màn hình, theo dạng sau:

ST T	HỌ VÀ TÊN	NGSIN H	NOI SINH	GHI CHÚ
1	Lê Văn	20/03/75	27 Nguyễn Huệ	_____
2	Hoàng	02/03/75	32 Lê Lợi	_____
.	Võ Viết Hòa
.

- Xử lý dữ liệu theo yêu cầu sau:

* Tính TONGDIEM = D_WIN + D_WORD + D_EXCEL

* Căn cứ vào TONGDIEM để xếp loại như sau:

Giỏi nếu TONGDIEM ≥ 24

Khá nếu 18 ≤ TONGDIEM < 24

T.Bình nếu 15 ≤ TONGDIEM < 18

Kém nếu TONGDIEM < 15

- In kết quả thi của các học viên ra màn hình theo dạng sau:

STT	HỌ VÀ TÊN	ĐIỂM	TỔNG	XẾP LOẠI
-----	-----------	------	------	----------

3. Viết lại chương trình trong bài tập 4, dùng danh sách liên kết, có bổ sung chức năng tìm kiếm sinh viên.

4. Hãy viết một hàm có hai đối số là hai con trỏ, mỗi con trỏ trỏ đến một danh sách liên kết, và nối hai danh sách lại với nhau, nối danh sách thứ hai sau danh sách thứ nhất.

5. Một stack là loại danh sách đặc biệt có các tính chất sau:

- Việc bổ sung phần tử được thực hiện ở cuối danh sách.
- Việc loại bỏ phần tử cũng được thực hiện ở cuối danh sách.

Viết chương trình minh họa các thao tác bổ sung (push) và loại bỏ (pop) trên stack.

6. Stack được gọi là hàng đợi theo kiểu vào trước ra sau (FILO - First In Last Out) bởi vì phần tử đầu tiên đẩy vào stack thì luôn luôn được lấy ra cuối cùng. Dùng push() và pop() và các hàm khác nếu cần, hãy viết một chương trình để đọc vào một dòng, xét xem nó có tính chất: khi đọc xuôi hay ngược đều cho ra cùng một kết quả không? Ví dụ: “Able was I ere I saw Elba”.

7. Viết chương trình minh họa việc thống kê các từ khóa của C được nhập vào từ bàn phím. Sử dụng thuật toán tìm kiếm từ khóa trên một mảng bằng phương pháp chia đôi.

8. Viết lại chương trình của bài tập 7, trong đó sử dụng con trỏ trong hàm tìm kiếm từ khóa trên một mảng.

BÀI THỰC HÀNH SỐ 8:: KIỂU TẬP TIN

A. CÂU HỎI

1. Sự khác nhau giữa stream văn bản và stream nhị phân là gì?
2. Chương trình của bạn phải làm gì trước khi nó có thể truy cập đến một tập tin?
3. Khi mở một tập tin với hàm fopen, cần phải chỉ định thông tin gì và hàm này trả về giá trị gì?
4. Hai phương thức tổng quát để đọc thông tin trong tập tin?
5. Giá trị EOF là gì? Khi nào EOF được dùng?
6. Làm thế nào để kiểm tra tình trạng của cuối tập tin trong mode văn bản và mode nhị phân?
7. Bộ định vị tập tin là gì? Làm thế nào có thể thay đổi nó?
8. Khi một tập tin được mở lần đầu tiên, bộ định vị tập tin trở vào đâu?
9. Viết mã để đóng tất cả stream tập tin.
10. Chỉ ra hai cách để di chuyển bộ định vị tập tin về đầu tập tin.

B. BÀI TẬP

1. Viết chương trình thực hiện các yêu cầu:
 - Nhập 10 số thực vào một tập tin văn bản có tên là INPUT.
 - Đọc nội dung tập tin INPUT.
 - Tính tổng bình phương các số có trong tập tin INPUT.
2. Viết chương trình thực hiện các yêu cầu:
 - Mở tập tin mới và nhập vào một số mẫu tin. Mỗi mẫu tin bao gồm các trường: họ tên, tuổi, lương. Quá trình nhập dữ liệu kết thúc khi họ tên nhập vào là rỗng.
 - Thêm dữ liệu vào tập tin.
 - Mở tập tin để đọc và hiển thị ra màn hình nội dung tập tin.
3. Tương tự bài 1, nhưng bổ sung các thao tác trên tập tin: tu sửa và thêm mẫu tin mới.
4. Mở một tập tin văn bản, tính kích thước và thống kê số lần xuất hiện các chữ cái trong tập tin văn bản đó.
5. Bài toán quản lý sinh viên. Viết chương trình thực hiện các yêu cầu:
 - Nhập dữ liệu các sinh viên vào một danh sách liên kết đơn, trường dữ liệu của mỗi nút bao gồm: Họ lót, tên, điểm toán, điểm tin. Sau đó lưu dữ liệu vào một tập tin có tên DULIEU.DAT
 - In danh sách sinh viên vừa nhập
 - Mở tập tin DULIEU.DAT để tu sửa các mẫu tin.
 - Đọc nội dung tập tin DULIEU.DAT vào danh sách liên kết đơn (sau khi đã tu sửa) và hiển thị ra màn hình các mẫu tin.
7. Viết chương trình in nội dung tập tin văn bản, chỉ in các ký tự chữ cái và chữ số. Tên tập tin là đối số của hàm main.
8. Viết một chương trình thực hiện các yêu cầu sau:
 - Cho phép soạn thảo văn bản trên DOS.
 - Các chức năng cơ bản: mở tập tin mới, mở tập tin đã có, lưu tập tin, thoát, trợ giúp.