



## Bài tập lớn Hiện thực một công cụ hỗ trợ tự động Giải bài toán (Solver)

### 1. Giới thiệu về Solver

Toán học ngày nay đã phát triển và được ứng dụng trong nhiều lĩnh vực. Solver được xem là một công cụ hoặc phần mềm được lập trình để hỗ trợ xác định lời giải (hoặc đáp án) cho một vấn đề toán học. Nó sẽ giúp giảm công sức tính toán cho người dùng trong một lĩnh vực chuyên môn đặc thù.

Một số solver điển hình được dùng thực tế hiện nay và khá nổi tiếng trong công nghiệp có thể liệt kê như sau:



**Excel Solver** là một công cụ có sẵn trong Excel, cung cấp các lệnh và các tính năng tùy chỉnh để giải quyết các vấn đề quyết định như các bài toán tối ưu, ví dụ như tối thiểu chi phí, tối đa lợi nhuận, ...



**COIN-OR** là một dự án phát triển một công cụ ứng dụng nhằm mục đích "tạo ra cho phần mềm toán học những gì tài liệu mở dành cho lý thuyết toán học"



**CPLEX Optimizer** là một phần mềm tối ưu được phát triển bởi IBM bằng cách sử dụng các giải thuật song song phân tán nhằm tận dụng nhiều máy tính để giải các bài toán khó.



**Gurobi Optimizer** là một phần mềm tối ưu chuyên dùng để giải các bài toán quy hoạch tuyến tính, quy hoạch toàn phương, ...

**Matlab** là phần mềm cung cấp môi trường tính toán số và lập trình, do công ty MathWorks thiết kế.

Và còn nhiều Solver nữa được áp dụng cho mỗi lĩnh vực đặc thù nhằm hướng đến ứng dụng nền tảng toán học và sự phát triển của lập trình để cung cấp những công cụ hữu ích chuyên dụng.

Trong khuôn khổ Bài tập lớn này, mỗi sinh viên cần hiện thực một công cụ Solver đơn giản có khả năng hỗ trợ tính toán các biểu thức số học (*arithmetic*) và các phép tính lượng giác (*trigonometry*).





## 2. Yêu cầu

Trong bài tập lớn này, sinh viên cần hiện thực một chương trình mô phỏng một solver đơn giản. Chương trình sẽ nhận thông tin đầu vào do người dùng nhập từ bàn phím bao gồm các thông số cấu hình cho phép tính cần thực hiện và giá trị đầu vào của phép tính. Sau đó, chương trình sẽ tính toán và in ra màn hình kết quả thu được sau quá trình thực hiện.

## 3. Mô tả chương trình

Dữ liệu đầu vào được người dùng nhập vào sẽ gồm nhiều dòng mà trong đó có dòng đầu tiên dùng để mô tả và các dòng kế tiếp là giá trị được nhập để tính toán.

Trong dòng đầu tiên gồm nhiều thông số cấu hình. Các thông số được cách nhau bởi 1 khoảng trắng. Format của dòng đầu tiên sẽ như sau:

<b>Menu_code</b> <Các thông số khác>
--------------------------------------

Thông số **Menu\_code** dùng để xác định Solver sẽ hỗ trợ tính toán trong ba phương pháp tính được nêu, cụ thể là:

- Nếu là 1: Phép tính lượng giác
- Nếu là 2: Phép tính số học

Ví dụ 1	
Input	Output
1 1 0 7 50	0.7660440
Đây là input cho <b>phép tính lượng giác</b> do Menu_code = 1.	

Ví dụ 2	
Input	Output
4	-1
Đây là input <b>không hợp lệ</b> do Menu_code = 4. Chương trình in ra -1.	



Các thông số khác sẽ khác nhau đối với từng phép tính và được mô tả cụ thể ở các mục **4, 5**.

## 4. Phép tính lượng giác

### 4.1. Dữ liệu đầu vào

Khi lựa chọn phép tính lượng giác, dữ liệu đầu vào gồm có 3 dòng. Dòng đầu tiên của dữ liệu đầu vào là **Menu\_code**, dòng thứ 2 sẽ gồm có **3** thông số cấu hình và dòng thứ 3 là giá trị cần tính toán. Dữ liệu đầu vào có format như sau:

1
n m p
x

Trong đó:

- **n**: xác định phép tính toán (sin/cos/tan)
- **m**: xác định đơn vị nhập vào (độ/radian)
- **p**: độ chính xác của kết quả tính toán.
- **x**: giá trị cần tính toán. Ví dụ: sin(x)

**Lưu ý:** Nếu có bất kì thông số cấu hình nào đó trong dòng số 2 không hợp lệ, chương trình in ra -1 và kết thúc (không cần phải nhập giá trị x).

Các thông số và dữ liệu nhập sẽ được mô tả chi tiết hơn bên dưới.

### 4.2. Dữ liệu xuất ra

Dữ liệu xuất ra sẽ là một số thực, là kết quả sau khi tính toán phép toán lượng giác và được làm tròn theo nguyên tắc của mục 4.5.

### 4.3. Thông số n

Thông số **n** là một số nguyên xác định phép tính toán tương ứng với 1 trong 3 giá trị:

- Nếu là 1: tính  $\sin(x)$
- Nếu là 2: tính  $\cos(x)$
- Nếu là 3: tính  $\tan(x)$



**Ví dụ 3**

Input	Output
1 2 1 4 0.6	0.8253
Đây là input cho phép tính lượng giác, thực hiện tính $\cos(x)$ do $n = 2$ .	

Nếu  $n$  nhận một giá trị khác thì đó là một giá trị không hợp lệ. Kết quả của việc nhập giá trị không hợp lệ là chương trình sẽ in ra -1 và kết thúc chương trình.

**Ví dụ 4**

Input	Output
1 7 0 2	-1
Đây là input cho phép tính lượng giác, nhưng có $n = 7$ là một <b>giá trị không hợp lệ</b> . Kết quả là chương trình sẽ in ra <b>-1</b> .	

Cách tính  $\sin(x)$ ,  $\cos(x)$ ,  $\tan(x)$  sẽ được mô tả cụ thể trong mục **4.7** và **4.8**.

#### 4.4. Thông số $m$

Thông số  $m$  là một số nguyên xác định góc nhập vào được tính theo độ hay radian.

- Nếu  $m$  là 0: góc được tính theo độ.
- Nếu  $m$  là 1: góc được tính theo radian.

**Ví dụ 5**

Input	Output
1 1 1 4 0.623	0.5835
Đây là input cho phép tính lượng giác, thực hiện tính $\sin(x)$ ( $n = 1$ ), với <b>góc <math>x</math> được tính theo radian</b> do thông số $m$ là 1.	



Nếu **m** nhận một giá trị khác 2 giá trị trên thì đó là một giá trị không hợp lệ, chương trình sẽ in ra giá trị **-1** và kết thúc.

Ví dụ 6	
Input	Output
1 3 2 7	-1
Đây là input <b>không hợp lệ</b> do thông số $m = 2$ . Chương trình in ra giá trị -1 và kết thúc.	

#### 4.5. Thông số **p**

Thông số **p** là một số nguyên xác định độ chính xác của kết quả tính toán, độ chính xác này là số chữ số của phần thập phân được lấy để hiển thị kết quả ra màn hình. Thông số **p** chỉ nhận giá trị là 1 trong 3 số nguyên sau:

- 2: kết quả tính toán được làm tròn đến 2 chữ số thập phân.
- 4: kết quả tính toán được làm tròn đến 4 chữ số thập phân.
- 7: kết quả tính toán được làm tròn đến 7 chữ số thập phân.

Ví dụ 7	
Input	Output
1 1 1 2 2.433	0.65
Đây là input cho phép tính lượng giác, thực hiện tính $\sin(x)$ ( $n = 1$ ), với đơn vị đo của $x$ là radian ( $m = 1$ ) và <b>kết quả tính được làm tròn đến 2 chữ số thập phân</b> do $p = 2$ .	

Nếu **p** nhận giá trị khác 3 giá trị trên thì đó là giá trị không hợp lệ. Kết quả của việc nhập giá trị không hợp lệ là chương trình sẽ in ra **-1** và kết thúc chương trình.



**Ví dụ 8**

Input	Output
1 1 0 5	-1
Đây là input <b>không hợp lệ</b> do thông số $p = 5$ . Chương trình in ra giá trị -1 và kết thúc.	

#### 4.6. Dữ liệu nhập x

Dữ liệu nhập x là một số thực tương ứng với góc nhập để thực hiện phép tính lượng giác. Để phép tính xấp xỉ được đúng, chương trình chỉ chấp nhận giá trị thực cho góc như sau:

- Nếu x tính theo độ:  $0 \leq x \leq 180$ .
- Nếu x tính theo radian:  $0 \leq x \leq \pi$ .

Riêng phép tính  $\tan(x)$  sẽ có trường hợp nhập x không hợp lệ, trường hợp này sẽ được mô tả trong phần tính  $\tan(x)$ .

**Ví dụ 9**

Input	Output
1 1 0 2 123	0.84
Đây là input cho phép tính lượng giác, thực hiện tính $\sin(x)$ ( $n = 1$ ), với đơn vị đo của x là độ ( $m = 0$ ), kết quả tính được làm tròn đến 2 chữ số thập phân ( $p = 2$ ) và <b>x = 123</b> .	



**Ví dụ 10**

Input	Output
1 3 1 7 4	-1
Đây là input <b>không hợp lệ</b> do dữ liệu $x = 4 > \pi$ . Chương trình in ra giá trị -1 và kết thúc.	

#### 4.7. Phép tính *sin*, *cos*

Công thức khai triển Taylor cho ta một đa thức xấp xỉ một hàm khả vi tại một điểm cho trước, chương trình sẽ dùng công thức này để tính toán xấp xỉ giá trị của hàm *sin*, *cos* với một góc được nhập vào.

Công thức xấp xỉ Taylor:

$$\sin(x) = \sum_{r=0}^{\infty} \frac{(-1)^r}{(2r+1)!} x^{2r+1} = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots$$

$$\cos(x) = \sum_{r=0}^{\infty} \frac{(-1)^r}{(2r)!} x^{2r} = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \dots$$

Tuy nhiên, với  $r > 5$  thì các phần tử được cộng thêm vào có giá trị khá nhỏ, chương trình sẽ tính *sin*, *cos* theo công thức xấp xỉ trên nhưng chỉ đến phần tử  $r = 5$ , cụ thể là

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

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

- Lưu ý về góc x trong công thức Taylor**

Đơn vị đo của góc x trong công thức Taylor là **radian**. Do đó, nếu góc nhập trong chương trình đang tính theo độ, cần phải **đổi góc này sang giá trị tương ứng theo radian** rồi mới thực hiện tính toán theo công thức Taylor. Công thức chuyển đổi góc từ độ sang radian:

$$x(^{\circ}) = x * \frac{\pi}{180} \text{ (rad)}$$

Trong đó, số  $\pi$  được lấy xấp xỉ bằng **3.14159**.



**Ví dụ 11**

Input	Output
1 1 1 4 0.489	0.4697
<p>Đây là input cho phép tính lượng giác:</p> <ul style="list-style-type: none"><li>Thực hiện tính <math>\sin(x)</math> (<math>n = 1</math>)</li><li>Đơn vị đo của <math>x</math> là radian (<math>m = 1</math>)</li><li>Kết quả tính được làm tròn đến 4 chữ số thập phân (<math>p = 4</math>).</li><li><math>x = 0.489</math></li></ul> <p>Vì đơn vị đo đã là radian, kết quả của phép tính trên là <math>\sin(0.489) = 0.4697</math></p>	

**Ví dụ 12**

Input	Output
1 2 0 2 123	-0.54
<p>Đây là input cho phép tính lượng giác:</p> <ul style="list-style-type: none"><li>Thực hiện tính <math>\cos(x)</math> (<math>n = 1</math>)</li><li>Đơn vị đo của <math>x</math> là độ (<math>m = 0</math>)</li><li>Kết quả tính được làm tròn đến 2 chữ số thập phân (<math>p = 2</math>).</li><li><math>x = 123</math></li></ul> <p>Vì đơn vị đo là độ, vì vậy nên cần biến đổi <math>123</math> (độ) <math>= 2.146754</math> (rad)</p> <p>Kết quả của phép tính trên là <math>\cos(2.146754) = 0.54</math></p>	





#### 4.8. Phép tính *tan*

Phép tính *tan* được tính như sau

$$\tan(x) = \frac{\sin(x)}{\cos(x)}$$

Với *sin*, *cos* được tính xấp xỉ theo mục 5. Do tính qua công thức xấp xỉ nên góc nhập *x* cũng phải được tính theo radian, nếu góc nhập đang tính theo độ thì đổi sang radian như mô tả ở mục 5.

- Lưu ý về góc không hợp lệ khi tính *tan*

*tan(x)* có thể không xác định khi *cos(x)* bằng 0. Do đó, *tan(x)* chỉ xác định khi:

- Nếu *x* tính theo độ:  $x \neq 90$
- Nếu *x* tính theo radian:  $x \neq \frac{\pi}{2}$

Nếu giá trị của *x* làm cho *tan(x)* không xác định thì giá trị đó được xem là không hợp lệ và chương trình sẽ in ra **-1**.

Ví dụ 13	
Input	Output
1 3 0 7 150	-0.5771655
<p>Đây là input cho phép tính lượng giác:</p> <ul style="list-style-type: none"><li>• Thực hiện tính <math>\tan(x)</math> (<math>n = 3</math>)</li><li>• Đơn vị đo của <math>x</math> là độ (<math>m = 0</math>)</li><li>• Kết quả tính được làm tròn đến 7 chữ số thập phân (<math>p = 7</math>).</li><li>• <math>x = 150</math></li></ul> <p>Vì đơn vị đo là độ, vì vậy nên cần biến đổi <math>150 (\text{độ}) = 2.61799 (\text{rad})</math></p> <p>Kết quả của phép tính trên là <math>\tan(2.61799) = -0.5771655</math></p>	



**Ví dụ 14**

Input	Output
1 3 0 2 90	-1
<p>Đây là input cho phép tính lượng giác:</p> <ul style="list-style-type: none"><li>Thực hiện tính <math>\tan(x)</math> (<math>m = 3</math>)</li><li>Đơn vị đo của <math>x</math> là độ (<math>n = 0</math>)</li><li>Kết quả tính được làm tròn đến 2 chữ số thập phân (<math>p = 2</math>).</li><li><math>x = 90</math></li></ul> <p>Vì <math>x = 90</math>, <math>m = 3</math> và <math>n = 0</math>, đây là phép tính <math>\tan(90)</math> với đơn vị đo là độ. Đây là một phép tính <b>không hợp lệ</b>. Chương trình in ra -1.</p>	

#### 4.9. Các phép tính lượng giác không hợp lệ

Nhắc lại một số trường hợp phép tính lượng giác không hợp lệ:

- Thông số  $n$  (phép tính toán) không phải là 1, 2, 3
- Thông số  $m$  (đơn vị đo) không phải là 0, 1
- Thông số  $p$  (làm tròn) không phải là 2, 4, 7
- Trong trường hợp  $n = 1$ :
  - Nếu  $m = 0$ , tức tính theo độ,  $x \neq 90$
  - Nếu  $m = 1$ , tức tính theo radian,  $x \neq \frac{\pi}{2}$



## **5. Phép tính số học**

### **5.1. Dữ liệu đầu vào**

Khi lựa chọn phép tính số học, đầu vào gồm có 3 dòng. Dòng đầu tiên của đầu vào là Menu\_code, dòng thứ 2 sẽ gồm có 1 thông số cấu hình và dòng thứ 3 là giá trị cần tính toán. Format của dữ liệu đầu vào như sau:

2
m
x

Trong đó:

- m: xác định phép toán (in các ước số/phân tích ra thừa số nguyên tố)
- x: giá trị cần tính

**Lưu ý:** Nếu có bất kì thông số cấu hình nào đó trong dòng số 2 không hợp lệ, chương trình in ra -1 và kết thúc (không cần phải nhập giá trị x).

Các thông số và dữ liệu nhập sẽ được mô tả chi tiết hơn bên dưới.

### **5.2. Dữ liệu xuất ra**

Dữ liệu xuất ra sẽ được mô tả cụ thể đối với từng phép tính số học và được trình bày cụ thể ở các mục 5.5 và 5.6.

### **5.3. Thông số m**

Thông số m là một số nguyên xác định phép tính toán tương ứng với 1 trong 2 giá trị sau:

- Nếu là 1: In các ước số của x
- Nếu là 2: Phân tích x ra tích các thừa số nguyên tố



**Ví dụ 15**

Input	Output
2 1 26	1 2 13 26
Đây là input cho phép tính số học, thực hiện việc <b>in các ước số</b> ( $m = 1$ ) của 26.	

Nếu  $m$  nhận một giá trị khác thì đó là một giá trị không hợp lệ. Kết quả của việc nhập giá trị không hợp lệ là chương trình sẽ in ra -1 và kết thúc chương trình.

**Ví dụ 16**

Input	Output
2 5	-1
Đây là input cho phép tính số học, nhưng có $m = 5$ là một giá trị <b>không hợp lệ</b> . Kết quả là chương trình sẽ in ra <b>-1</b> .	

#### 5.4. Dữ liệu nhập vào $x$

Dữ liệu nhập  $x$  là một **số nguyên lớn hơn 1** tương ứng với số cần tính toán số học. Mọi trường hợp nhập vào số  $x \leq 1$  được xem là một giá trị không hợp lệ.

**Ví dụ 17**

Input	Output
2 2 25	$5^2$
Đây là input cho phép tính số học, thực hiện phân tích ra thừa số nguyên tố ( $m = 2$ ) cho số $x = 25$ .	



**Ví dụ 18**

Input	Output
2 2 -223	-1
Đây là input cho phép tính số học, thực hiện phân tích ra thừa số nguyên tố ( $m = 2$ ) cho số -223 ( $x = -223$ ). Vì $x \leq 1$ , đây là một giá trị không hợp lệ, chương trình in ra <b>-1</b> .	

### 5.5. In các ước của một số

Chương trình thực hiện việc in tất cả các ước số của  $x$  (tức các số mà  $x$  chia hết cho).

Với các yêu cầu như sau:

- Các ước số được in từ nhỏ tới lớn.
- Mỗi ước số cách nhau bởi một khoảng trắng.
- Kết thúc việc in không được phép có thêm khoảng trắng.
- 1 và  $x$  cũng được tính là ước số của  $x$ .

**Ví dụ 19**

Input	Output
2 1 24	1 2 3 4 6 8 12 24
Đây là input cho phép tính số học, thực hiện việc in các ước số ( $m = 1$ ) của 26 ( $x = 26$ ). Chương trình in ra: 1 2 3 4 6 8 12 24<Không dư khoảng trắng>	



## 5.6. Phân tích ra các thừa số nguyên tố

Chương trình thực hiện việc phân tích số  $x$  ra các thừa số nguyên tố. Format của của việc in ra được mô tả như sau:

$$d1^{p1} * d2^{p2} * d3^{p3} \dots$$

Trong đó:

- $d1, d2, d3, \dots, dn$ : Các thừa số nguyên tố
- $p1, p2, p3, \dots, pn$ : Số mũ của các thừa số nguyên tố
- $^$ : Toán tử lũy thừa
- $*$ : Toán tử nhân

Một số lưu ý:

- Thừa số nguyên tố sẽ được in nếu số mũ của nó **lớn hơn 0**.
- Các thừa số nguyên tố được in ra từ nhỏ tới lớn.
- Các thừa số nguyên tố và số mũ đều là số nguyên dương.

Ví dụ 20	
Input	Output
2 2 30240	$2^5 * 3^3 * 5^1 * 7^1$
Đây là input cho phép tính số học, thực hiện việc phân tích ra thừa số nguyên tố ( $m = 2$ ) của 30240 ( $x = 30240$ ). Chương trình in ra: $2^5 * 3^3 * 5^1 * 7^1$ <Không dư khoảng trắng>	



**Ví dụ 21**

Input	Output
2 2 79	$79^1$
<p>Đây là input cho phép tính số học, thực hiện việc phân tích ra thừa số nguyên tố (<math>m = 2</math>) của 79 (<math>x = 79</math>).</p> <p>Chương trình in ra:</p> <p><math>79^1</math></p>	

**Ví dụ 22**

Input	Output
2 2 1	-1
<p>Đây là input cho phép tính số học, thực hiện việc phân tích ra thừa số nguyên tố (<math>m = 2</math>) của 1 (<math>x = 1</math>). Vì <math>x \leq 1</math> nên đây là một phép tính không hợp lệ</p> <p>Chương trình in ra:</p> <p><b>-1</b></p>	



## 6. Một số yêu cầu và hướng dẫn khác

- Sinh viên không được phép dùng thư viện `<math.h>`, `<cmath>` hay các thư viện liên quan có hiện thực sẵn các hàm tính toán sin/cos/tan.
- Vì kết quả liên quan đến độ chính xác, sinh viên phải dùng kiểu dữ liệu **double** cho các biến số thực để đảm bảo độ chính xác.
- Mã nguồn khởi tạo đã hiện thực sẵn hàm `exitInvalid()`. Mã nguồn hiện tại đã hoạt động đúng cho trường hợp không hợp lệ đối với thông số cấu hình **Menu\_code**. Sinh viên tham khảo mã nguồn khởi tạo và nên sử dụng hàm `exitInvalid()` để giải quyết cho các trường hợp không hợp lệ.

## 7. Nộp bài

Sinh viên download file [NMLT\\_assignment.zip](#) từ trang Web của môn học. Khi giải nén file này, sẽ có được các file sau:

<code>solver.c</code>	Chương trình khởi tạo
<code>[NMLT]SolverAssignment.pdf</code>	File mô tả nội dung bài tập lớn

File `solver.c` là mã nguồn khởi tạo, trong đó các hàm `exitInvalid()` đã được viết sẵn để giải quyết các trường hợp không hợp lệ. Sinh viên được khuyến khích sử dụng lại hàm này.

Khi nộp bài, sinh viên nộp bài trên site e-Learning của môn học. Sinh viên điền code bài tập lớn giống như các bài thực hành khác. Sinh viên được cung cấp 3 nơi nộp bài:

- Phần 1 - Phần Lượng giác (**Bài tập lớn – Phần Lượng giác**): Sinh viên nộp bài làm và được chấm trên **30 testcases** để kiểm tra các trường hợp liên quan đến phần Lượng giác.
- Phần 2 - Phần Số học (**Bài tập lớn – Phần Số học**): Sinh viên nộp bài làm và được chấm trên **20 testcases** để kiểm tra các trường hợp liên quan đến phần Số học.
- Phần 3 - Phần Khó (**Bài tập lớn – Phần Khó**): Phần này gồm có 50 testcases kiểm tra cả phần Lượng giác và phần Số học, trong đó có một vài testcases khó. Sinh viên cần đạt tối thiểu **7 điểm** ở cả 2 phần trên mới có thể được nộp bài vào





**TRƯỜNG ĐẠI HỌC BÁCH KHOA TP.HCM**  
**Khoa Khoa học & Kỹ thuật Máy tính**

Phần Khó. Sinh viên nếu không nộp bài vào Phần Khó sẽ được tính là **0 điểm** cho phần này.

Trong mỗi phần trên, sinh viên có tối đa **10 lần** làm bài. Đối với mỗi lần làm bài, sinh viên có **10 phút** để nộp code và kiểm tra. Chỉ có lần nhấn "Kiểm tra" đầu tiên là được tính điểm, các lần sau sẽ không được lấy điểm. Kết quả bài làm chỉ hiển thị sau khi bạn nhấn nút "Hoàn thành bài làm". Điểm cao nhất trong các lần làm bài sẽ được lấy làm điểm cho phần đó.

Điểm bài tập lớn của sinh viên được tính theo công thức sau:

$$\text{Điểm BTL} = (\text{Điểm phần 1} * 0.6 + \text{Điểm phần 2} * 0.4) * 0.7 + \text{Điểm phần 3} * 0.3$$

Ví dụ, sinh viên A đạt 7 điểm ở Phần 1 và 8 điểm ở Phần 2. Sinh viên A đủ điều kiện để nộp Phần 3, ở phần này kết quả A đạt được là 6 điểm. Vậy:

$$\text{Điểm BTL của A} = (7 * 0.6 + 8 * 0.4) * 0.7 + 6 * 0.3 = 6.98$$

**Sinh viên phải kiểm tra chương trình của mình trên MinGW trước khi nộp.**

### **7.1. Thời hạn nộp bài**

Thời hạn chót để nộp bài là **23h55 chủ nhật ngày 25/04/2021**. Sinh viên nộp bài trên site e-Learning của môn học. Nếu sinh viên nộp bài trễ, điểm tối đa của bài tập lớn sẽ là **0 điểm**. KHÔNG nhận bài được gửi qua mail hoặc bất kỳ hình thức nào khác.

### **7.2. Xử lý gian lận**

Bài tập lớn phải được sinh viên tự làm. Sinh viên sẽ bị coi là gian lận nếu:

- Có sự giống nhau bất thường giữa mã nguồn của các bài nộp. Trong trường hợp này, tất cả các bài nộp đều bị coi là gian lận. Do vậy sinh viên phải bảo vệ mã nguồn bài tập lớn của mình.
- Sinh viên không hiểu mã nguồn do chính mình viết, trừ những phần mã được cung cấp sẵn trong chương trình khởi tạo. Sinh viên có thể tham khảo từ bất kỳ nguồn tài liệu nào, tuy nhiên phải đảm bảo rằng mình hiểu rõ ý nghĩa của tất cả những dòng lệnh mà mình viết. Trong trường hợp không hiểu rõ mã nguồn của nơi mình tham khảo, sinh viên được đặc biệt cảnh báo là không được sử dụng mã nguồn này; thay vào đó nên sử dụng những gì đã được học để viết chương trình.



**TRƯỜNG ĐẠI HỌC BÁCH KHOA TP.HCM**  
**Khoa Khoa học & Kỹ thuật Máy tính**

Trong trường hợp bị kết luận là gian lận, sinh viên sẽ bị điểm 0 cho toàn bộ môn học (không chỉ bài tập lớn).

**KHÔNG CHẤP NHẬN BẤT KỲ GIẢI THÍCH NÀO VÀ KHÔNG CÓ BẤT KỲ NGOẠI LỆ NÀO!**

Sau mỗi bài tập lớn được nộp, sẽ có một số sinh viên được gọi phỏng vấn ngẫu nhiên để chứng minh rằng bài tập lớn vừa được nộp là do chính mình làm.

**-HẾT-**