

Bài 2. Thuật toán và các cấu trúc điều khiển

Mục tiêu: *Luyện tập các cấu trúc điều khiển*

Sử dụng các cấu trúc điều khiển để cài đặt một số thuật toán đơn giản.

Tự thiết kế thuật toán và cài đặt cho một số bài toán đơn giản

*Yêu cầu làm ít nhất 05 bài phần C hoặc 01 bài (***).*

A. Thực hành

1. Luyện các cấu trúc lặp. Hãy viết vòng lặp in ra chuỗi các số chẵn dương không vượt quá $n = 100$ bằng vòng for, vòng while, vòng do-while bằng càng nhiều cách càng tốt. gợi ý các khả năng để kết hợp thành các cách làm khác nhau. Cách nào bạn cho là tốt nhất (theo tiêu chí gọn gàng đơn giản)?
 - a. dùng con đếm xác định số lần lặp
 - b. không xác định số lần lặp
 - c. chỉ dùng điều kiện kết thúc của cấu trúc điều khiển, không dùng continue/break
 - d. dùng break
 - e. dùng continue
2. Tương tự với bài toán cho một số nguyên k , hãy thực hiện phép tính $k = k/3$ cho đến khi k nhỏ hơn 3. In ra giá trị cuối cùng của k .
3. Nếu n nhập từ bàn phím thay vì biết trước thì trong hai bài trên có những vòng lặp nào bạn đã biết không còn chạy đúng nữa? Nếu cần thì bạn hãy chạy thử với các loại giá trị của n như dương, bằng 0, âm.

4. Luyện tập các cấu trúc rẽ nhánh:

Viết chương trình nhập vào một ngày (date) ở dạng dd-mm-yyyy.

Tiếp theo là một loạt các số nguyên, mỗi số trên một dòng.

Với mỗi số k , chương trình cần in ra trên một dòng kết quả là ngày (date dạng dd-mm-yyyy) đã nhập cộng thêm k ngày (day). Input kết thúc khi gặp $k = 0$.

Ví dụ Input (5 dòng, 0 đánh dấu kết thúc):

22-03-2015

1

-1

10

0

Output (3 dòng chứa 3 kết quả):

23-03-2015

21-03-2015

01-04-2015

Yêu cầu: Tự viết các test để kiểm tra tính đúng đắn của chương trình.

Gợi ý: có những trường hợp nào có thể xảy ra? tháng Hai, năm nhuận, cộng trừ trong tháng, cộng trừ vượt ranh giới 1 tháng, cộng trừ vượt ranh giới 2 tháng, cộng trừ vượt ranh giới tháng 31 ngày, tháng 30 ngày, tháng 28 ngày, tháng 29 ngày, cộng trừ bên trong phạm vi năm, cộng trừ vượt ranh giới năm, cộng trừ vượt ranh giới năm nhuận....

5. (nếu không tự làm được thì gv thực hành sẽ hướng dẫn lại) Chạy chương trình với input output chuẩn được điều hướng từ file. Việc này để tiết kiệm công sức chạy chương trình, giúp chạy được nhiều test.

Môi trường: nếu là máy linux thì đã xong, nếu là windows và đang dùng gnu trong bộ mingw hoặc trong devC++ thì kiểm tra xem có chạy được lệnh g++ ở ngoài cửa sổ dòng lệnh hay không. Nếu không thì đi tìm xem trình biên dịch gcc nằm ở đâu và đặt đường dẫn path đến đó.

Cuối cùng thì tại **cmd** cần chạy được các lệnh như sau:

để chỉ biên dịch/compile, chưa ra được file exe

g++ -c hello.cpp

hoặc để dịch thẳng ra file tên hello (compile + build)

g++ -o hello hello.cpp

Sau khi đã build được, bạn sẽ có thể chạy file hello.exe là kết quả compile và build

Đến đây bạn có thể định hướng input từ file và output ra file mà không động đến mã nguồn.

Giả sử hello.cpp dùng input/output chuẩn, chẳng hạn dùng cin để đọc và cout để ghi, nghĩa là bình thường bạn chạy chương trình và dùng bàn phím để nhập dữ liệu.

Bây giờ bạn có thể gõ toàn bộ input vào một file, chẳng hạn in.txt và chạy lệnh sau:

hello < in.txt

chương trình của bạn sẽ đọc từ in.txt và in kết quả ra màn hình thay vì đợi bạn gõ.

còn nếu dùng lệnh

hello < in.txt > out.txt

chương trình của bạn sẽ đọc từ in.txt và đổ kết quả vào file out.txt thay vì đợi bạn gõ và in ra màn hình.

Tất nhiên, nếu bạn chạy lệnh hello không có định hướng <> thì chương trình vẫn đợi bạn nhập input như bình thường.

B. Câu hỏi

Câu 1. Đoạn mã nào phía dưới có thể thay thế cho lệnh (tam nguyên) sau:

min = (a < b) ? a : b;

a. if (a > b) min = a;

else min = b;

b. if (a < b) min = a;

else min = b;

c. (a > b) min = a;

else min = b;

- d. Không có đáp án đúng

Câu 2. Khoanh tròn đáp án đúng:

- a. Nhãn **default** là bắt buộc phải có trong cấu trúc lệnh **switch**.
- b. Lệnh ngắt **break** là bắt buộc có trong tập lệnh mặc định của nhãn **default** trong cấu trúc lệnh **switch**.
- c. Quên lệnh ngắt **break** trong cấu trúc lệnh **switch** là mắc lỗi biên dịch.
- d. Ứng dụng hay gặp của cấu trúc lệnh **switch** là tạo MENU.

Câu 3. Các nhãn **case** trong cấu trúc **switch** phải là:

- a. Biểu thức trả về kiểu số nguyên
- b. Biểu thức trả về kiểu ký tự
- c. Phải là hằng (kiểu số nguyên hoặc kiểu ký tự), chứ không thể là biến hay biểu thức.
- d. Kiểu số thực

Câu 4. Khoanh tròn đáp án đúng cho đoạn mã lệnh sau:

```
int n = 5, k;  
for (k = 1; k <= 50; k++) {  
    if ( (k%n)) continue;  
    cout << k << "\t";  
}
```

- a. In ra các số nguyên không chia hết cho 5 nằm trong khoảng từ 1 tới 50.
- b. In ra các số nguyên chia hết cho 5 nằm trong khoảng từ 1 tới 50.
- c. In ra các số nguyên nằm trong khoảng từ 1 tới 50.
- d. Không in ra gì vì biểu thức logic trong phần điều kiện của lệnh **if** sai.

Câu 5. Trong cấu trúc của vòng lặp **for** (*khởi_tạo; điều_kiện_lặp; cập_nhật*)

- a. Nhất định phải đủ cả ba biểu thức *khởi_tạo*, *điều_kiện_lặp* và *cập_nhật*.
- b. Có thể thiếu hai trong ba biểu thức trên
- c. Có thể thiếu một trong ba biểu thức trên
- d. Có thể thiếu bất cứ biểu thức nào trong ba biểu thức trên.

Câu 6. Để kết thúc sự lặp lại hiện hành mà không thực thi các câu lệnh còn lại trong khối vòng lặp. Sự điều khiển trở về ngay lập tức với điều kiện kiểm tra, xác định xem có cần kết thúc chính vòng lặp hoặc tiếp tục với sự lặp lại kế tiếp hay không, ta gọi lệnh nào sau đây:

- a. **break**;
- b. **continue**;
- c. **goto**;
- d. **return (0)**;

Câu 7. Khoanh tròn đáp án đúng cho đoạn mã lệnh sau:

```
int x = 10;  
for (int x = 0; x <= 20; x ++);
```

```
cout << x;
```

- a. In ra 20
- b. In ra 21
- c. In ra 10
- d. Các đáp án trên đều sai, đoạn mã lệnh có lỗi

Câu 8. Khoanh tròn **các** đáp án **sai**:

- a. Trong 3 kiểu lặp: while, do- while và for thì **do- while** là linh hoạt nhất.
- b. Trong 3 kiểu lặp: while, do- while và for thì **while** là linh hoạt nhất
- c. Trong 3 kiểu lặp: while, do- while và for thì **for** là linh hoạt nhất
- d. Một đoạn code bất kỳ sử dụng một trong 3 kiểu lặp **while, do- while** hoặc **for** thì nó cũng có thể được xây dựng bằng một trong hai kiểu lặp còn lại.
- e. **for** là phép lặp “đếm” tự nhiên.

Câu 9. Cho đoạn chương trình sau:

```
if (i == j) {  
    cout << "A" << endl;  
}  
else if ((i % j) < 3) {  
    cout << "B" << endl;  
}  
else if (i < (j-1)) {  
    cout << "C" << endl;  
}  
else {  
    cout << "D" << endl;  
}
```

Nếu $i = 4$ và $j = 9$, kết quả xuất ra màn hình sẽ là?

- a. A
- b. D
- c. B
- d. C

Câu 10. Cho đoạn chương trình sau:

```
#include <iostream>  
using namespace std;  
  
int main()  
{  
    int x=7,y=45,z=63,min_num;  
    if (x<z)  
        min_num=x;  
    else
```

```

        min_num=z;
    if(y<z)
        min_num=y;
    else
        min_num=z;

    cout<<"The minimum is:" <<min_num;
    system ("pause");
    return 0;
}

```

- a. 7
- b. 45
- c. 63
- d. Tất cả các đáp trên đều sai

Câu 11. Kết quả xuất ra khi biên dịch khối lệnh sau:

```

int num = 5, sum = 0;
while (num > 0)
{
    if (num %2 !=0)
        sum+=num;
    --num;
}
cout << "The sum is " << sum << endl;

```

- a. The sum is 0
- b. The sum is 5
- c. The sum is 9
- d. The sum is 15

Câu 12. Cho biết kết quả xuất ra của đoạn chương trình sau đây trên màn hình là gì?

```

int1 = 120;
cin >> int2;          // Giả sử người sử dụng nhập vào 30
if ((int1 > 100) && (int2 = 50))
int3 = int1 + int2;
else
int3 = int1 - int2;
cout << int1 << ' ' << int2 << ' ' << int3;

```

- a. 120 30 150
- b. 120 50 170
- c. 120 30 170

d. 120 30 90

Câu 13. Cho biết kết quả xuất ra của đoạn chương trình sau đây trên màn hình là gì?

```
int a=1,b=2;
if (a<b)
    if (a>b) a=a+1;
    else    b=b+1;
else      a=a+b;
cout<<a*b;
```

- a. 2
- b. 3
- c. 6
- d. 12

Câu 14. Trong các đoạn code phía dưới, đoạn nào không có lỗi?

a. while (c <= 5)
{
 product *= c;
 c++;

b. if (gender == 1)
 cout << "Woman" << endl;
 else;
 cout << "Man" << endl;

c. while (z >= 0) // nguoi dung khoi tao z = 2
 sum += z;

d. while (i <= y)
{
 power *= x;
 i++;
}

Câu 15. Khoanh tròn đáp án đúng cho đoạn code bên dưới:

```
int x = -1;
do {
    cout<<"Hello, world!\t";
    x++;
} while ( !x );
```

In ra màn hình:

- a. Hello, world!
- b. Hello, world! Hello, world!

- c. Hello, world! Hello, world! Hello, world!
- d. Không in gì

Câu 16. Khoanh tròn đáp án đúng cho đoạn code bên dưới:

```
int x = -3;
do {
    cout<<"Hello, world!\t";
    x++;
} while ( x );
```

In ra màn hình:

- a. Hello, world!
- b. Hello, world! Hello, world!
- c. Hello, world! Hello, world! Hello, world!
- d. Không in gì

Câu 17. Khoanh tròn đáp án đúng cho đoạn code bên dưới:

```
int x = -3;
while (x) {
    cout<<"Hello, world!\t";
    x++;
}
```

In ra màn hình:

- a. Hello, world!
- b. Hello, world! Hello, world!
- c. Hello, world! Hello, world! Hello, world!
- d. Không in gì

Câu 18. Khoanh tròn đáp án đúng cho đoạn code bên dưới:

```
int x = -3;
while (!x) {
    cout<<"Hello, world!\t";
    x++;
}
```

In ra màn hình:

- a. Hello, world!
- b. Hello, world! Hello, world!
- c. Hello, world! Hello, world! Hello, world!
- d. Không in gì

Câu 19. Khoanh tròn đáp án đúng cho đoạn code bên dưới:

```
int x = -3;
while (-3) {
    cout<<"Hello, world!\t";
```

```

        x++;
        if (!x)
            break;
    }

```

In ra màn hình:

- Hello, world!
- Hello, world! Hello, world!
- Hello, world! Hello, world! Hello, world!
- Không in gì

Câu 20. Khoanh tròn đáp án đúng cho đoạn code bên dưới:

```

int i = -3;
for( ; ; ) {
    cout<<"Hello, world!\n";
    i ++;
    if (i == 0) break;
};

```

In ra màn hình:

- Hello, world! Hello, world!
- Hello, world! Hello, world! Hello, world!
- Đoạn code có lỗi.
- Không in gì.

C. Bài tập

- Viết chương trình Prime đọc vào một số và in ra 'yes' nếu đó là một số nguyên tố. Nếu không thì in ra 'no'.
- Viết chương trình nhập một số nguyên dương N và in ra một tam giác kích thước N. Chẳng hạn với N bằng 5 thì tam giác có dạng như sau:

```

*****
****
***
**
*

```

- Tương tự bài trên với dạng tam giác sau (với N bằng 5):

```

*
***
*****
*****

```

4. Tương tự bài trên với dạng tam giác sau (với N bằng 5):

*

5. Viết chương trình NumberToEnglish với nhiệm vụ đọc từ bàn phím một số trong khoảng từ -999.999.999 tới 999.999.999 và in ra cách đọc số đó bằng tiếng Anh. Bạn cần dùng đến các từ: negative, zero, one, two, three, four, five, six, seven, eight, nine, ten, eleven, twelve, thirteen, fourteen, fifteen, sixteen, seventeen, eighteen, nineteen, twenty, thirty, forty, fifty, sixty, seventy, eighty, ninety, hundred, thousand, million.

6. Viết chương trình nhập vào 1 số n ($n < 100$) và in ra ma trận $n \times n$ dạng như sau.

Ví dụ:input

10

output:

```
1 2 3 4 5 6 7 8 9 10
2 3 4 5 6 7 8 9 10 1
3 4 5 6 7 8 9 10 1 2
4 5 6 7 8 9 10 1 2 3
5 6 7 8 9 10 1 2 3 4
```

...

7. (*) Viết chương trình DistinctNumbers với nhiệm vụ đọc một chuỗi số từ input chuẩn (bàn phím) kết thúc ở số âm đầu tiên, và in ra các số đó theo đúng thứ tự, nhưng nếu có các số liền nhau cùng giá trị thì chỉ in ra một lần. Chương trình chỉ được dùng các biến kiểu cơ bản, không được dùng mảng, string, hay các cấu trúc dữ liệu khác.

Ví dụ input:

1 2 2 2 2 4 2 3 5 5 -1

Output tương ứng:

1 2 4 2 3 5 -1

8. (***) Viết chương trình BeautifulStrings với nhiệm vụ đọc một chuỗi kí tự là chữ cái tiếng Anh từ input chuẩn (bàn phím) kết thúc khi hết input, nhiệm vụ in ra số dãy con khác nhau sinh ra từ việc cắt đi hai kí tự bất kì trong chuỗi kí tự đầu vào. Không giới hạn kích thước input, không dùng string, mảng hay các cấu trúc dữ liệu không cơ bản, chỉ dùng các cấu trúc từ đầu môn học đến giờ là đủ.

Nguồn: <https://www.hackerrank.com/contests/101hack34/challenges/beautiful-string/>

(các bạn vào đó xem chi tiết và chạy test nhé :))

9. (**) Rectangular Game. Cho một lưới vuông hai chiều vô tận với ô ở góc trái dưới có tọa độ (1,1). Tất cả các ô ban đầu đều có giá trị bằng 0. Hãy chơi một trò chơi.

Trò chơi gồm N bước. Mỗi bước, bạn được cho một cặp số nguyên a và b. Giá trị của mỗi ô tọa độ (u,v) thỏa mãn $1 \leq u \leq a$ và $1 \leq v \leq b$ đều được tăng thêm 1 đơn vị. Sau N bước như vậy, nếu X là giá trị lớn nhất trong tất cả các ô. Bạn hãy cho biết có bao nhiêu số X trong lưới?

Input:

Dòng đầu tiên của input chứa số N. N dòng tiếp theo, mỗi dòng chứa hai số nguyên a và b, cách nhau bởi một dấu cách.

Output:

Ghi ra một số kiểu int là số số X

Ràng buộc:

$1 \leq N \leq 100$

$1 \leq a \leq 1000000$

$1 \leq b \leq 1000000$

Sample input:	Sample output:
3 2 3 3 7 4 1	2

Nguồn <https://www.hackerrank.com/challenges/rectangular-game> (chạy test tại đó nhé)

Còn tiếp....

(thầy Khôi sẽ bổ sung vào đây)

Bài 1. Tìm lỗi trong mỗi vòng lặp sau:

1. `while (n<20);`
 `cout << n++ <<endl;`
2. `for (int i = 1, i <= 8, ++i)`
 `cout << 1.0/i << endl;`
3. `int n = 10;`
 `do`
 `cout << 1.0/n ;`
 `n++;`
 `while (n < 20);`
4. `for (int i=10; i<20;) {`
 `cout << i*i <<endl;`
 `i--;`
}

Bài 2. Nhập vào một số thực x và một số nguyên y. Tính x mũ y rồi cho hiển thị lên màn hình. Kiểm tra lại kết quả bằng hàm pow (x , y).

Bài 3. Viết chương trình nhập vào từ bàn phím ba số. Kiểm tra xem 3 số đó có thỏa mãn là độ dài các cạnh của một tam giác hay không (áp dụng bất đẳng thức tam giác)? Nếu có, in ra màn hình thông báo về chu vi của tam giác đó và cho biết tam giác đó là loại tam giác gì (đều, cân, vuông, ...).

Bài 4. Nhập hai số nguyên dương x, y. In bội số chung nhỏ nhất.

Bài 5. Nhập hai số nguyên dương x, y. In ước số chung lớn nhất.

Bài 6. Nhập tử số và mẫu số của một phân số, in ra phân số tối giản.

Bài 7. Nhập vào 6 số là các hệ số của hệ phương trình 2 ẩn bậc một. Kiểm tra xem hệ phương trình có nghiệm hay không, có vô số nghiệm hay không, là những nghiệm nào.

Bài 8. Viết phương trình nhập vào 3 hệ số của một phương trình bậc hai. Giải phương trình bậc hai đó, có tính nghiệm phức.

Bài 9. Viết chương trình tính tổng các chữ số của 1 số nguyên dương được nhập vào từ bàn phím.

Bài 10. Để chuyển đổi nhiệt độ dưới dạng độ F (Fahrenheit) sang nhiệt độ dưới dạng độ C (Celsius), ta trừ cho 32, nhân 5 sau đó chia cho 9. Để chuyển đổi nhiệt độ dưới dạng độ C (Celsius) sang nhiệt độ dưới dạng độ K (Kelvin), ta cộng thêm 273.15. Viết chương trình hiển thị ra màn hình giá trị tương ứng độ C, độ K của một số nguyên nhập vào từ bàn phím. Định dạng in ra như trong màn hình dưới:

Fahrenheit	Celsius	Absolute Value
0	-17.78	255.37
20	-6.67	266.48
40	4.44	277.59
...
...
300	148.89	422.04

Bài 11. Nhập vào ba số Min, Max và Step_size. Viết chương trình chuyển đổi nhiệt độ (như trong bài 4) với nhiệt độ F bắt đầu từ Min, kết thúc ở Max và nhảy theo Step_size. Trong bài 9 Min = 0, Max = 300 và Step_size = 20.

Bài 12. Viết chương trình nhập số nguyên dương n, kiểm tra n có là số chính phương không?

Bài 13. Nhập số N, in ra các số nguyên tố, các số hoàn hảo (hoàn chỉnh), các số chính phương từ 2...N.

Bài 14. Tìm giá trị lớn nhất, nhỏ nhất trong 1 dãy có n số thực nhập vào từ bàn phím (không được dùng mảng).

Bài 15. Nhập số N, kiểm tra N có thuộc dãy Fibonaxi không? In ra màn hình các số thuộc dãy Fibonaxi $\leq N$.

Bài 16. Viết chương trình nhập một chữ số từ màn hình, in ra thông báo chữ số đó dưới dạng chữ. Đọc các chữ số đến khi nào chữ số nhập nằm ngoài khoảng 0..9. Nếu giá trị nhập vào nằm ngoài khoảng 0..9 thì dừng chương trình.

Kết quả ví dụ:

```
Nhap vao mot so trong khoang 0..9: 0
Vua nhap chu so khong
```

Nhap vao mot so trong khoang 0..9: 5

Vua nhap chu so nam

Nhap vao mot so trong khoang 0..9: 10

Gợi ý: sử dụng cấu trúc switch. Sử dụng cấu trúc do ... while để thực hiện yêu cầu lặp. (<biểu thức điều kiện>) sẽ kiểm tra số nhập vào có nằm trong khoảng từ 0 đến 9 hay không.

Bài 17. Viết chương trình nhập vào điểm số 5 môn học. Sau đó in ra điểm trung bình cộng của 5 môn đó.

Bài 18. Viết chương trình tính điểm trung bình cho các môn học. Chương trình cho phép nhập vào điểm số của từng môn học trong khoảng từ 0 ... 10. Nếu điểm số nhập vào lớn hơn 10 thì yêu cầu nhập lại. Nếu điểm số nhỏ hơn 0 thì in ra điểm trung bình và kết thúc chương trình.

Kết quả ví dụ:

Diem mon hoc 1: 7

Diem mon hoc 2: 11

Nhap lai diem mon hoc 2 (trong khoang tu 0..10): 1

Diem mon hoc 3: -5

So mon hoc: 2

Diem trung binh cua 2 mon hoc: 4

Bài 19. Viết chương trình tính điểm trung bình cho các môn học. Số môn học được nhập vào từ bàn phím. Với mỗi môn học có điểm số (từ 0,0..10,0) và hệ số (1 1,5 2 2,5 hoặc 3). Viết chương trình in ra tổng số môn học, tổng số hệ số và điểm trung bình. Chương trình cần kiểm tra các điều kiện cần thiết.

Kết quả ví dụ:

Tong so mon hoc can tinh DTB: 2

Diem mon hoc 1: 7

He so mon hoc 1: 1

Diem mon hoc 2: 3

He so mon hoc 2: 3

So mon hoc: 2

Tong so he so: 4

Diem trung binh cua 2 mon hoc: 4

Bài 20. Bài tập lớn

Phần A. Một người lao động được trả lương hàng tháng qua tài khoản ngân hàng vào đầu tháng cho khối lượng công việc họ hoàn thành vào tháng trước (ví dụ đầu tháng 2 nhận được lương của tháng 1). Hợp đồng làm tối thiểu mỗi tháng là 100 tiếng với mức lương 12.000đ/giờ. Nếu trong tháng người đó làm vượt thời gian 100 tiếng thì mức lương theo giờ được tính như sau:

Thời gian làm quá 100 tiếng	Mức lương theo giờ (đồng/giờ)
-----------------------------	-------------------------------

1-50 tiếng	16.000
------------	--------

1-50 tiếng tiếp theo	20.000
----------------------	--------

Từ những tiếng tiếp theo	25.000
--------------------------	--------

Ví dụ người đó nhận được lương mỗi tháng như sau:

$$TN(210 \text{ tiếng}) = 100 \cdot 12.000 + 50 \cdot 16.000 + 50 \cdot 20.000 + 10 \cdot 25.000 = 3.250.000\text{đ}$$

$$TN(167 \text{ tiếng}) = 100 \cdot 12.000 + 50 \cdot 16.000 + 17 \cdot 20.000 = 2.340.000\text{đ}$$

Viết chương trình nhận đầu vào là tổng số giờ làm việc một tháng, đầu ra là lương của tháng đó để giải quyết bài toán tính thu nhập của người lao động miêu tả như trên.

Phần B. Thu nhập của người lao động sẽ bị áp thuế. Mức tính thuế như sau:

Thu nhập mỗi tháng	Mức thuế phải chịu
--------------------	--------------------

1.000.000đ đầu tiên	0%
---------------------	----

500.000đ tiếp theo	10%
--------------------	-----

500.000đ tiếp theo	15%
--------------------	-----

Tiếp theo	20%
-----------	-----

Tuy nhiên người lao động còn phải đóng 03 loại bảo hiểm: xã hội, y tế, nghề nghiệp với mức bảo hiểm là 9% trên tổng thu nhập hàng tháng. Toàn bộ tiền đóng bảo hiểm được miễn trừ thuế.

Ví dụ thu nhập là 3.000.000đ/tháng thì thuế phải đóng là:

$$\text{Bảo_Hiểm}(9\%) = 3.000.000 \cdot 9\% = 270.000\text{đ}$$

$$TN_chịu_thuế = \text{Tổng_TN} - \text{Bảo_Hiểm}(9\%) = 3.000.000 - 270.000 = 2.730.000\text{đ}$$

$$\text{Thuế} = 1.000.000 \cdot 0\% + 500.000 \cdot 10\% + 500.000 \cdot 15\% + 730.000 \cdot 20\% = 271.000\text{đ}$$

$$TN_sau_thuế = \text{Tổng_TN} - \text{Bảo_Hiểm} - \text{Thuế} = 2.459.000\text{đ}$$

Viết chương trình để tính thuế người lao động phải trả và số tiền họ nhận được sau thuế mỗi tháng.

Phần C. Người lao động có nhu cầu vay tiêu dùng trong vòng 1 năm. Lãi suất vay là 2%/tháng được tính trên tổng dư nợ và cộng dồn vào cuối mỗi tháng.

Ví dụ, với khoản vay 1.000.000đ, sau một năm phải trả 1.268.243đ.

Chú ý: số tiền được làm tròn đến đơn vị đồng sau mỗi tháng.

Tháng	Tổng dư nợ đầu tháng	Lãi suất vay	Tổng dư nợ cuối tháng
1	1.000.000	20.000	1.020.000
2	1.020.000	20.400	1.040.400
3	1.040.400	20.808	1.061.208
4	1.061.208	21.224	1.082.432
5	1.082.432	21.649	1.104.081
6	1.104.081	22.082	1.126.163
7	1.126.163	22.523	1.148.686
8	1.148.686	22.974	1.171.660
9	1.171.660	23.433	1.195.093
10	1.195.093	23.902	1.218.995
11	1.218.995	24.380	1.243.375
12	1.243.375	24.868	1.268.243

Viết chương trình để tính tổng dư nợ của một khoản vay ngân hàng trong vòng 01 năm.

Phần D. Một sinh viên vào đầu năm học quyết định vay ngân hàng 10.000.000đ để mua

laptop. Ngân hàng đồng ý cho vay với lãi suất 2%/tháng và yêu cầu phải trả ngân hàng một khoản tiền vào đầu mỗi tháng. Tổng thu nhập của sinh viên bao gồm trợ cấp từ bố mẹ 1.500.000đ/tháng vào đầu tháng và tiền lương từ việc làm thêm (như miêu tả trong **Phần A** và **Phần B**). Sinh viên đó dự tính sẽ cần nhiều nhất 2.000.000đ cho sinh hoạt hàng tháng và số tiền còn lại được trả cho ngân hàng.

Viết chương trình để tính sau bao nhiêu tháng sinh viên đó hoàn trả hết nợ (gốc và lãi) cho ngân hàng được biết mỗi tháng sinh viên đó làm trung bình 155 tiếng.

Chương trình sẽ in ra thu nhập sau thuế của sinh viên đó từ việc làm thêm, số tiền lãi ngân hàng tính vào cuối mỗi tháng và số tiền dư nợ của sinh viên vào đầu tháng (sau khi thanh toán hàng tháng cho ngân hàng)

Phần E. Sửa chương trình trong **Phần D** để giải quyết trường hợp thời gian làm việc của sinh viên đó trong mỗi tháng là khác nhau:

	Tổng số giờ
Tháng 09/2016	155
Tháng 10/2016	145
Tháng 11/2016	125
Tháng 12/2016	135
Tháng 01/2017	100
Tháng 02/2017	100
Tháng 03/2017	145
Tháng 04/2017	135
Tháng 05/2017	100
Tháng 06/2017	155
Tháng 07/2017	170
Tháng 08/2017	180

Được biết sinh viên đó được ngân hàng quyết định vay vào đầu tháng 9/2016.

Viết chương trình để tính sau bao nhiêu tháng sinh viên đó hoàn trả hết nợ (gốc và lãi) cho ngân hàng.

Chương trình sẽ in ra:

- thu nhập sau thuế của sinh viên đó từ việc làm thêm
- số tiền lãi ngân hàng tính vào cuối mỗi tháng
- số tiền dư nợ của sinh viên vào đầu tháng (sau khi thanh toán hàng tháng cho ngân hàng)
- sau bao nhiêu tháng thì sinh viên đó thanh toán hết
- khoản thanh toán cuối cùng với ngân hàng