

# BÀI TẬP 1

## CHUYÊN ĐỀ TỔ CHỨC DỮ LIỆU

Họ và tên: Bùi phúc kiến

MSSV: 20880034

**Câu 1. (1.5.8)** Nhập một số nguyên và xuất ra số nguyên đó với dấu phẩy (,) phân cách mỗi 3 chữ số.

**Bài làm:**

**Ý tưởng:**

- Người dùng nhập số nguyên không dấu x (4 bytes) từ 0 đến 4294967295.
- Dùng mảng a[10] có kích thước bằng 10 để lưu các phần tử có được từ việc tách x thành dãy gồm số có 3 chữ số từ sau ra trước qua các bước lặp với phép toán  $x \% 1000$  và  $x = x / 1000$ .

Ví dụ: x = 1023456789 thì a[0] = 789, a[1] = 456, a[2] = 23, a[3] = 1.

- In các phần tử của mảng a theo thứ tự từ sau ra trước.

**Code:**

```
//Dùng để in các phần tử ở giữa của mảng, phần tử có 2 chữ số thì thêm số 0 ở trước, phần tử có 1 chữ số thì thêm 00 ở trước.
void printThreeDigits(int x)
{
    if(x >= 100)
    {
        printf("%d", x);
    }
    else if(x >= 10 && x < 100)
    {
        printf("0%d", x);
    }
    else
    {
        printf("00%d", x);
    }
}

// Xuất số nguyên với các dấu phẩy
void outPutIntegerWithComma(int x)
{
    int a[10];
    int count = 0;
    while(x != 0)
    {
        a[count] = x%1000;
        count++;
        x = x/1000;
    }
    //In phần tử đầu
    printf("%d, ", a[count-1]);

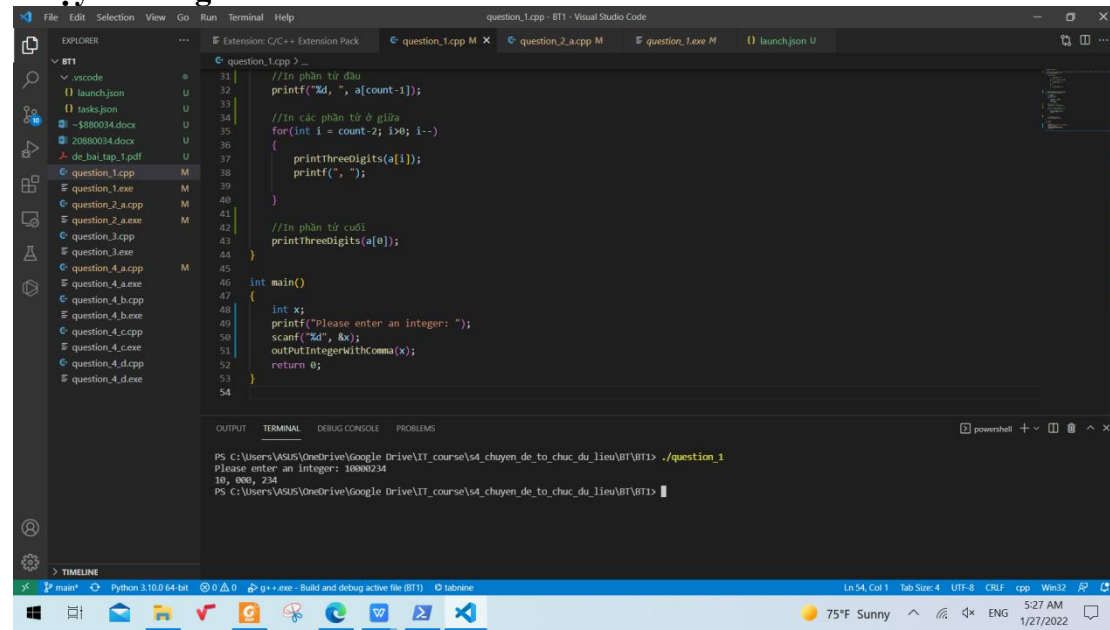
    //In các phần tử ở giữa
    for(int i = count-2; i>0; i--)
    {
        printThreeDigits(a[i]);
        printf(", ");
    }
    //In phần tử cuối
    printThreeDigits(a[0]);
}
```

```

int main()
{
    int x;
    printf("Please enter an integer: ");
    scanf("%d", &x);
    outPutIntegerWithComma(x);
    return 0;
}

```

## Chạy chương trình:



## Câu 2.

a) (1.5.10) Viết hàm tính căn bậc 3 của một số thực dương dùng phương pháp chia đôi.

**Bài làm:**

**Ý tưởng:** Với số thực dương  $y$ ,  $\sqrt[3]{y}$  sẽ rơi vào khoảng  $(0;1)$  nếu  $y < 1$  hoặc  $(0;y)$  nếu  $y > 1$ . Do đó ta dùng các biến left, right tương ứng 2 đầu mút của khoảng tìm kiếm. Biến  $mid = (left + right)/2$ , nếu  $mid^3 > y$  ta sẽ thu hẹp khoảng tìm kiếm bằng cách loại bỏ nửa khoảng bên trái (gán  $left = mid$ ), ngược lại ta sẽ loại bỏ nửa khoảng bên phải (gán  $right = mid$ ). Lặp lại cho đến khi khoảng tìm kiếm đủ hẹp (đạt độ chính xác như yêu cầu) thì dừng.

**Code:**

```
double myCubeRoot(double y)
{
    double mid, left, right;
    left = 0;
    if(y > 1)
    {
        right = y;
    }
    else
    {
        right = 1;
    }

    do
    {
        mid = (left + right)/2;
        if(mid*mid*mid > y)
        {
            right = mid;
        }
        else
        {
            left = mid;
        }
    }
    while((right - left)/right > 0.0000001);

    return mid;
}
```

b) Viết hàm tính căn bậc 3 của một số thực dương dùng phương pháp Newton.

**Bài làm:**

**Ý tưởng:** Với số thực  $m$ , giả sử  $x_0 = \sqrt[3]{m}$  khi đó  $x_0^3 = m \Leftrightarrow x_0^3 - m = 0$ . Đặt  $f(x) = x^3 - m$  thì ta có  $x_0$  là nghiệm của  $f(x)$ . Theo phương pháp Newton, dãy  $x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} = x_n - \frac{x_n^3 - m}{3x_n^2}$  với  $x_1$  là giá trị tùy ý (chọn  $x_1 = 1$ ) sẽ hội tụ về  $x_0$ . Khi  $n$  đủ lớn, ta có  $x_n$  xấp xỉ căn bậc ba của  $m$ .

**Code:**

```
double myCubeRoot(double m)
{
    double x = 1;
    do{
        x = x - (x*x*x - m)/(3*x*x);
    }
    while(abs(x*x*x - m)>0.000000000001);
    return x;
}

int main()
{
    for(int i = -10; i < 10; i++){
        double x = 0.1*i;
        double k = x*x*x;
        double r = myCubeRoot(k);
        printf("%f\t%f\n", k, r);
    }
    return 0;
}
```

## Chạy chương trình:

The screenshot shows the Visual Studio Code interface with the following details:

- Explorer Panel:** Displays a project named 'BT1' containing files like 'launch.json', 'tasks.json', '20880034.docx', 'de\_hai\_tap\_1.pdf', and several 'question\_\*.cpp' and 'question\_\*.exe' files.
- Main Editor:** Shows the source code of 'question\_2.b.cpp'. The code defines a function 'myCubeRoot' and a 'main' function that iterates from -10 to 10, calculating the cube root of x\*x\*x.
- Output Panel:** Displays the program's output, showing a table of values for k and r. The output is as follows:
 

k	r
-1.000000	-1.000000
-0.729000	-0.900000
-0.512000	-0.800000
-0.343000	-0.700000
-0.216000	-0.600000
-0.125000	-0.500000
-0.064000	-0.400000
-0.027000	-0.300000
-0.008000	-0.200000
-0.001000	-0.100000
0.000000	0.000000
0.001000	0.100000
0.008000	0.200000
0.027000	0.300000
0.064000	0.400000
0.125000	0.500000
0.216000	0.600000
0.343000	0.700000
0.512000	0.800000
0.729000	0.900000
1.000000	1.000000
- Status Bar:** Shows 'Ln 22, Col 17', 'Tab Size: 4', 'UTF-8', 'CRLF', 'cpp', 'Win32', and the system clock '9:32 AM 08-Feb-22'.

**Câu 3.** Xuất một trích đoạn Truyện Kiều tự chọn có dấu tiếng Việt (tương tự bài tập 1.7.6)

a) Trình bày giải pháp.

b) Viết mã.

c) Chụp màn hình kết quả.

**Bài làm:**

**a) Trình bày giải pháp:**

Trước tiên ta đặt lược đồ mã cho cửa sổ console là UTF-8 bằng cách dùng hàm SetConsoleOutputCP trong windows.h (thực hiện trên máy với hệ điều hành window) với đối số là 65001 (Mã số của lược đồ mã UTF-8). Sau đó chọn font cho cửa sổ console là Consolas - font unicode có hỗ trợ các ký tự tiếng Việt.

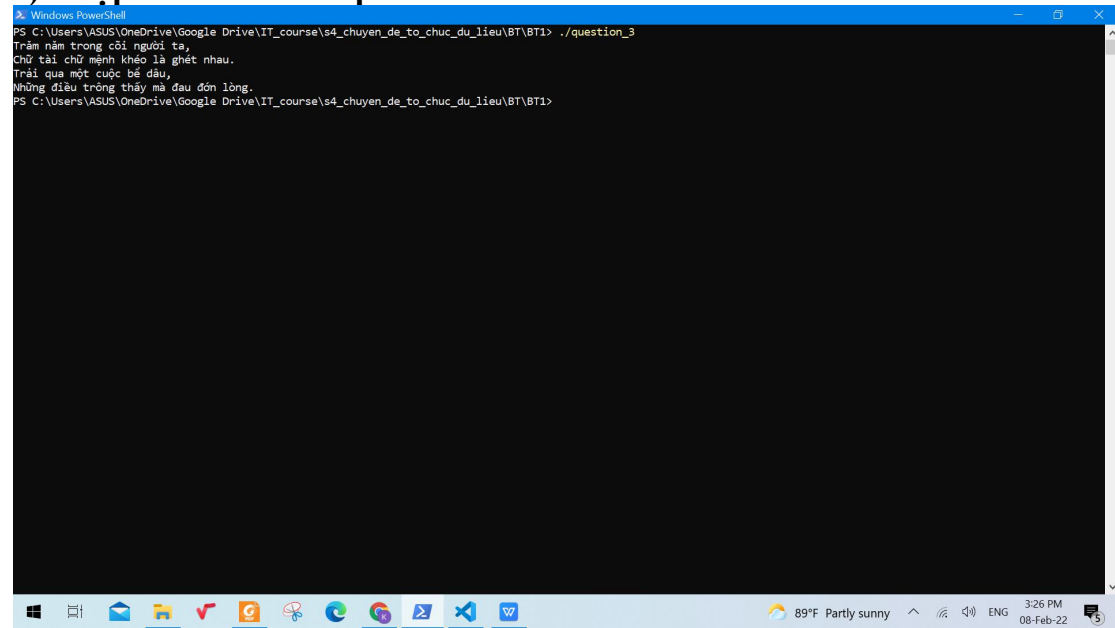
Các câu trong đoạn trích từ Truyện Kiều được chuyển sang mã unicode sử dụng website sau: <https://r12a.github.io/app-conversion/>. Các mã unicode có được được thêm vào '\x' bằng một chương trình viết bằng Python.

```
f = open("truyen_kieu.txt", 'r')
str = f.read()
str_list = str.split(' ')
new_str = ''
for e in str_list:
    new_str += '\\x' + 'x' + e
f1 = open('tk.txt', 'w')
f1.write(new_str)
```

**b) Viết mã.**

```
#include <stdio.h>
#include <windows.h>
int main()
{
    SetConsoleOutputCP(65001);
    printf("\x54\x72\xC4\x83\x6D\x20\x6E\xC4\x83\x6D\x20\x74\x72\x6F\x6E\x67\x20\x63\xC3\xB5\x69\x20\x6E\x67\xC6\xB0\xE1\xBB\x9D\x69\x20\x74\x61\x2C\x0A\x43\x68\xE1\xBB\xAF\x20\x74\xC3\xA0\x69\x20\x63\x68\xE1\xBB\xAF\x20\x6D\xE1\xBB\x87\x6E\x68\x20\x6B\x68\xC3\xA9\x6F\x20\x6C\xC3\xA0\x20\x67\x68\xC3\xA9\x74\x20\x6E\x68\x61\x75\x2E\x0A\x54\x72\xE1\xBA\xA3\x69\x20\x71\x75\x61\x20\x6D\xE1\xBB\x99\x74\x20\x63\x75\xE1\xBB\x99\x63\x20\x62\xE1\xBB\x83\x20\x64\xC3\xA2\x75\x2C\x0A\x4E\x68\xE1\xBB\xAF\x6E\x67\x20\xC4\x91\x69\xE1\xBB\x81\x75\x20\x74\x72\xC3\xB4\x6E\x67\x20\x74\x68\xE1\xBA\xA5\x79\x20\x6D\xC3\xA0\x20\xC4\x91\x61\x75\x20\xC4\x91\xE1\xBB\x9B\x6E\x20\x6C\xC3\xB2\x6E\x67\x2E");
    return 0;
}
```

### c) Chụp màn hình kết quả.



The screenshot shows a Windows PowerShell window with a blue title bar. The command prompt displays the following text:

```
PS C:\Users\ASUS\OneDrive\Google Drive\IT_course\s4_chuyen_de_to_chuc_du_lieu\BT\BT1> ../question_3
```

The output of the command is a Vietnamese poem:

```
Trăm năm trong cõi người ta,  
Chữ tài chữ mệnh khéo là ghét nhau.  
Trải qua một cuộc bể dâu,  
Những điều trông thấy mà đau đớn lòng.
```

The command prompt then shows the current directory:

```
PS C:\Users\ASUS\OneDrive\Google Drive\IT_course\s4_chuyen_de_to_chuc_du_lieu\BT\BT1>
```

The Windows taskbar is visible at the bottom, showing various application icons and system status information: 89°F Partly sunny, 3:26 PM, 08-Feb-22.

**Câu 4.** (Bài tập 2.1.1) Viết chương trình nhập giá trị thực  $x$  và tính giá trị thực của biểu thức sau bằng cách chỉ dùng các toán tử phù hợp (mà không dùng hàm hay các cấu trúc chọn, lặp)

a)  $x^{20}$

b)  $4x^{64} + 3x^{36} + 2(x-1)^8 + 1$

c)  $|x|$

d)  $sign(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x = 0 \\ -1 & \text{if } x < 0 \end{cases}$

**Bài làm:**

a) Tính  $x^{20}$ : bằng việc sử dụng các biến tạm, ta có thể tính  $x^{20}$  qua 5 phép toán nhân.

**Code:**

```
float raisedTo20(float x){
    float y = x*x;
    y = y*y; // y = x^4
    float z = y*y;
    z = z*z; // z = x^16
    return y*z; //(x^4)*(x^16) = x^20
}

int main(){
    float x;
    printf("Please enter a number: ");
    scanf("%f", &x);
    float y = raisedTo20(x);
    printf("\n%f raised to 20 is %f", x, y);
    return 0;
}
```

b) Tính  $4x^{64} + 3x^{36} + 2(x-1)^8 + 1$

```
double square(double x){
    return x*x;
}

double raisedTo4(double x){
    double y = square(x);
    return y*y;
}
```

```
double raisedTo8(double x){
    double y = raisedTo4(x);
    return y*y;
}
```

```
double raisedTo16(double x){
    double y = raisedTo8(x);
    return y*y;
}
```

```
double raisedTo32(double x){
    double y = raisedTo16(x);
    return y*y;
}
```

```
double raisedTo36(double x){
    double z = raisedTo4(x); //z = x^4
    double y = raisedTo8(z); //y = z^8 = (x^4)^8 = x^32
    return y*z;             //(x^32)*(x^4) = x^36
}
```

```
double raisedTo64(double x){
    double y = raisedTo32(x);
    return y*y;
}
```

```
double evaluateThePolynomial(double x){
    double y = raisedTo32(x); // y = x^32
    double z = y*y;           // z = y^2 = x^64
    double t = y*raisedTo4(x); // t = y*(x^4) = (x^32)*(x^4) = x^36
    double w = raisedTo8(x-1); // w = (x-1)^8
    double result = 4*z + 3*t + 2*w + 1;
    return result;
}
```

```
int main(){
    double x;
    printf("Please enter a number: ");
    scanf("%lf", &x);
    double y = evaluateThePolynomial(x);
    printf("\nThe value of the polynomial is %f",y);
    return 0;
}
```

$$d) \text{ sign}(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x = 0 \\ -1 & \text{if } x < 0 \end{cases}$$

Ta có các toán tử logic  $(x > 0) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x \leq 0 \end{cases}$  và  $(x < 0) = \begin{cases} 1 & \text{if } x < 0 \\ 0 & \text{if } x \geq 0 \end{cases}$  nên biểu

$$\text{thức } (x > 0) - (x < 0) = \begin{cases} 1 - 0 = 1 & \text{if } x > 0 \\ 0 - 0 = 0 & \text{if } x = 0 \\ 0 - 1 = -1 & \text{if } x < 0 \end{cases} = \text{sign}(x)$$

**Code:**

```
int sign(double x){
    return (x>0) - (x<0);
}

int main(){
    double x;
    printf("Please enter a number: ");
    scanf("%lf", &x);
    int y = sign(x);
    printf("\nsign(%f) = %d", x, y);
    return 0;
}
```



c)  $|x|$

Ta có  $|x| = x * \text{sign}(x) = x * [(x > 0) - (x < 0)]$

**Code:**

```
double absoluteValue(double x){
    return x*((x>0)-(x<0));
}

int main(){
    double x;
    printf("Please enter a number: ");
    scanf("%lf", &x);
    double y = absoluteValue(x);
    printf("The absolute value of %f is %f", x, y);
    return 0;
}
```