

<b>Trạng thái</b>	Đã xong
<b>Bắt đầu vào lúc</b>	Thứ Bảy, 4 tháng 5 2024, 2:13 AM
<b>Kết thúc lúc</b>	Chủ Nhật, 5 tháng 5 2024, 10:33 AM
<b>Thời gian thực hiện</b>	1 ngày 8 giờ



## Câu hỏi 1

Đúng

Đạt điểm 1,00

**[Tiếng Việt]**

Một chuỗi được gọi là palindrome nếu chuỗi đó giống với chuỗi được đảo ngược từ chính nó. Ví dụ: "eye", "noon", "abcbA"...

Hãy viết hàm kiểm tra xem một chuỗi có là palindrome hay không?

**Đầu vào:**

- `const char* str`: chuỗi cần kiểm tra palindrome. `str` chỉ bao gồm chữ cái thường

**Đầu ra:**

- `bool`: true nếu chuỗi `str` là palindrome, ngược lại false

**[English]**

A string is a palindrome if it reads the same forward and backward. For example: "eye", "noon", "abcbA", ...

Write a function to check if a given string is a palindrome

**Input:**

- `const char* str`: the string to be checked. `str` only contains lowercase letters

**Output:**

- `bool`: true if `str` is a palindrome, false otherwise

**For example:**

Test	Result
<pre>const char* str = "abba"; cout &lt;&lt; isPalindrome(str);</pre>	1
<pre>const char* str = "axz"; cout &lt;&lt; isPalindrome(str);</pre>	0

**Answer:** (penalty regime: 0 %)

Reset answer

```
1 bool isPalindrome(const std::string& str) {  
2     int left = 0;  
3     int right = str.length() - 1;  
4  
5     while (left < right) {  
6         if (str[left] != str[right]) {  
7             return false; // Không là palindrome  
8         }  
9         ++left;  
10        --right;  
11    }  
12  
13    return true; // Là palindrome  
14 }  
15
```



	Test	Expected	Got	
✓	<pre>const char* str = "abba"; cout &lt;&lt; isPalindrome(str);</pre>	1	1	✓
✓	<pre>const char* str = "axz"; cout &lt;&lt; isPalindrome(str);</pre>	0	0	✓

Passed all tests! ✓

## Câu hỏi 2

Đúng

Đạt điểm 1,00

**[Tiếng Việt]**

Một số tự nhiên  $n$  được gọi là đặc biệt khi và chỉ khi  $n$  là số nguyên tố và tổng các chữ số của  $n$  cũng là số nguyên tố. Viết hàm kiểm tra một số tự nhiên có đặc biệt hay không.

**Đầu vào:**

- int  $n$ : số tự nhiên cần kiểm tra có phải số đặc biệt không

**Đầu ra:**

- bool: trả về true nếu  $n$  là số đặc biệt, ngược lại trả về false

**[English]**

A natural number  $n$  is special if and only if  $n$  is a prime number and the sum of all the digits of  $n$  is also a prime number. Write a function that determines if a natural number is a special or not.

**Input:**

int  $n$ : a natural number  $n$ .  $0 \leq n \leq 1000$

**Output:**

bool: return true if  $n$  is special, return false otherwise

**For example:**

Test	Input	Result
<pre>int n; cin &gt;&gt; n; cout &lt;&lt; isSpecialNumber(n);</pre>	23	1
<pre>int n; cin &gt;&gt; n; cout &lt;&lt; isSpecialNumber(n);</pre>	7	1

**Answer:** (penalty regime: 0 %)



[Reset answer](#)

```
1 bool isPrime(int num) {
2     if (num < 2) {
3         return false;
4     }
5     for (int i = 2; i*i <= num; ++i) {
6         if (num % i == 0) {
7             return false;
8         }
9     }
10    return true;
11 }
12
13 // Hàm kiểm tra xem một số có phải là số đặc biệt hay không
14 bool isSpecialNumber(int n) {
15     if (!isPrime(n)) {
16         return false; // Không phải số nguyên tố
17     }
18
19     int digitSum = 0;
20     int temp = n;
21     while (temp > 0) {
22         digitSum += temp % 10;
23         temp /= 10;
24     }
25
26     return isPrime(digitSum);
27 }
28
```



	Test	Input	Expected	Got	
✓	<pre>int n; cin &gt;&gt; n; cout &lt;&lt; isSpecialNumber(n);</pre>	23	1	1	✓
✓	<pre>int n; cin &gt;&gt; n; cout &lt;&lt; isSpecialNumber(n);</pre>	7	1	1	✓

Passed all tests! ✓

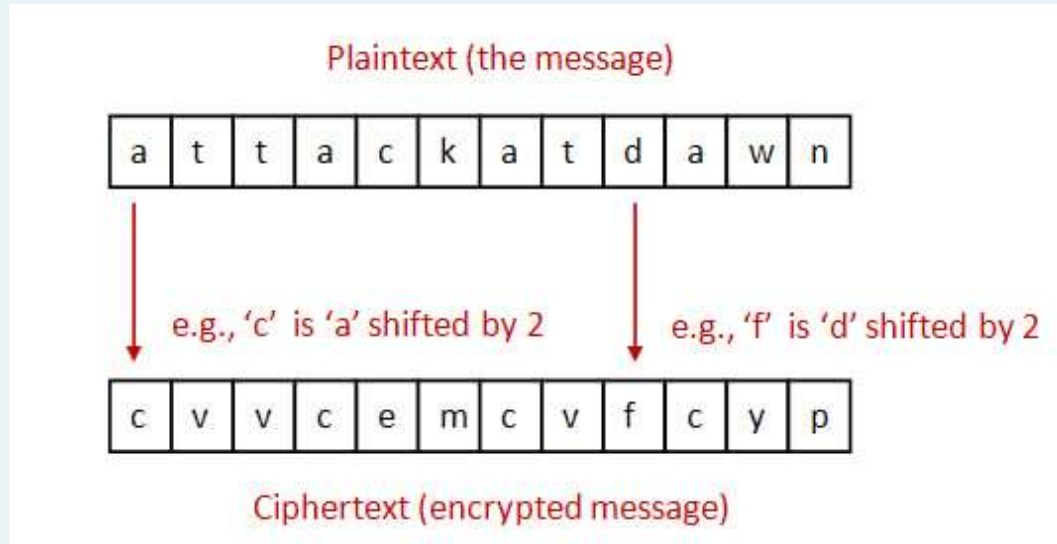
## Câu hỏi 3

Đúng một phần

Đạt điểm 1,00

**[Tiếng Việt]**

Viết một hàm mã hóa và một hàm giải mã một đoạn text theo phương pháp Caesar Cipher. Để mã hoá và giải mã một chuỗi ký tự text, ta cần một tham số có giá trị nguyên là shift.



Hàm mã hóa (tên **encrypt**) sẽ thay đổi từng chữ cái trong text bằng cách dịch chuyển chữ cái đó sang phải shift lần trong bảng chữ cái. Ví dụ với shift = 3. Khi đó 'a' được mã hoá thành 'd', 'b' được mã hoá thành 'e',... 'z' được mã hoá thành 'c'.

Hàm giải mã (tên **decrypt**) sẽ nhận một chuỗi ký tự text và giá trị nguyên shift và giải mã chuỗi ký tự này thành chuỗi ban đầu (tức dịch chuyển từng chữ cái sang trái shift lần trong bảng chữ cái)

**Đầu vào:**

- **char\* text:** chuỗi ký tự cần được mã hoá hoặc giải mã, chỉ bao gồm chữ cái thường và hoa
- **int shift:** giá trị dịch chuyển trong Caesar Cipher

**Đầu ra:**

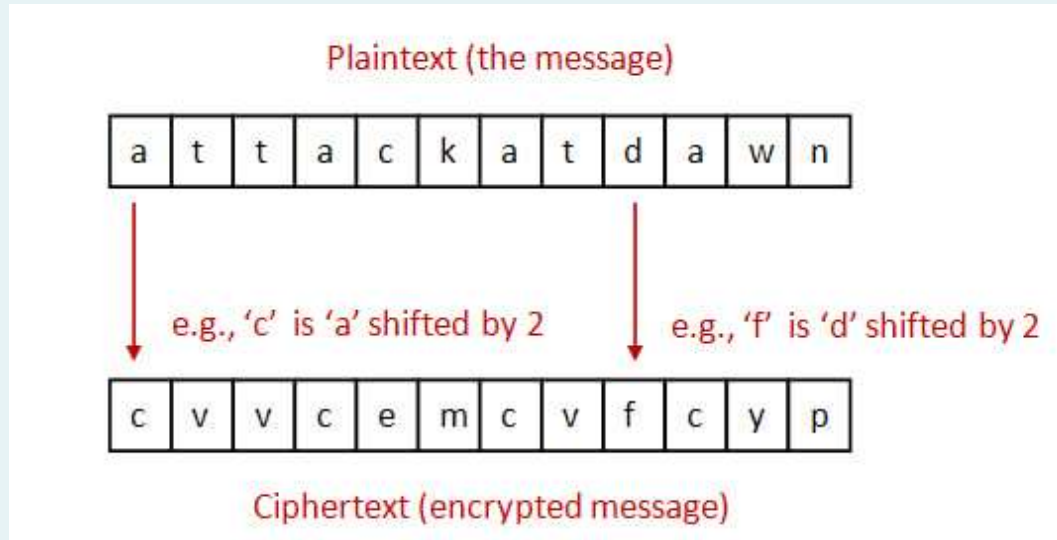
- Hàm không trả về.
- Chuỗi ký tự truyền vào **text** sẽ thay đổi trực tiếp trong hàm.

**[English]**

Write a function to encrypt and a function to decrypt a text string using Caesar Cipher technique.



In this technique, to encrypt a string we need a parameter of type integer called 'shift'.



The **encrypt** method will shift each letter by some fixed number of position (determined by the parameter 'shift') to the right in the alphabet. For example, when 'shift' is 3, 'a' will be replaced by 'd', 'b' will become 'e', ... , 'z' will become 'c'.

The **decrypt** method will receive an encoded string and a shift value and it will decode this string to get the original string, which means shifting each character to the left in the alphabet.

#### Input:

- **char\* text:** the text string that needs to be encrypted or decrypted. text only contains lowercase and uppercase ASCII letters
- **int shift:** the shift value in Caesar Cipher technique

#### Output:

- The function returns nothing.
- The input parameter **text** will be updated in-place.

#### For example:



Test	Input	Result
<pre>int n, shift; cin &gt;&gt; n &gt;&gt; shift; char* text = new char[n+1]; for(int i = 0; i &lt; n; i++) cin &gt;&gt; text[i]; text[n] = 0;  encrypt(text, shift); cout &lt;&lt; text &lt;&lt; '\n'; decrypt(text, shift); cout &lt;&lt; text;  delete[] text;</pre>	<pre>6 3 aczDYZ</pre>	<pre>dfcGBC aczDYZ</pre>
<pre>int n, shift; cin &gt;&gt; n &gt;&gt; shift; char* text = new char[n+1]; for(int i = 0; i &lt; n; i++) cin &gt;&gt; text[i]; text[n] = 0;  encrypt(text, shift); cout &lt;&lt; text &lt;&lt; '\n'; decrypt(text, shift); cout &lt;&lt; text;  delete[] text;</pre>	<pre>16 25 programmingisfun</pre>	<pre>oqnfqzllhmfhretm programmingisfun</pre>

**Answer:** (penalty regime: 0 %)

Reset answer



1	void encrypt(char* text, int shift) {	
2	for (int i = 0; text[i] != '\0'; i++) {	
3	if (isalpha(text[i])) {	
4	char base = islower(text[i]) ? 'a' : 'A';	
5	text[i] = (text[i] - base + shift) % 26 + base;	
6	}	

```
6      }
7    }
8  }
9
10 void decrypt(char* text, int shift) {
11     for (int i = 0; text[i] != '\0'; i++) {
12         if (isalpha(text[i])) {
13             char base = islower(text[i]) ? 'a' : 'A';
14             text[i] = (text[i] - base - shift + 26) % 26 + base; // Adding 26 to handle negati
15         }
16     }
17 }
18
```



	Test	Input	Expected	Got	
✓	<pre>int n, shift; cin &gt;&gt; n &gt;&gt; shift; char* text = new char[n+1]; for(int i = 0; i &lt; n; i++) cin &gt;&gt; text[i]; text[n] = 0;  encrypt(text, shift); cout &lt;&lt; text &lt;&lt; '\n'; decrypt(text, shift); cout &lt;&lt; text;  delete[] text;</pre>	<pre>6 3 aczDYZ</pre>	<pre>dfcGBC aczDYZ</pre>	<pre>dfcGBC aczDYZ</pre>	✓



	Test	Input	Expected	Got	
✓	<pre>int n, shift; cin &gt;&gt; n &gt;&gt; shift; char* text = new char[n+1]; for(int i = 0; i &lt; n; i++) cin &gt;&gt; text[i]; text[n] = 0;  encrypt(text, shift); cout &lt;&lt; text &lt;&lt; '\n'; decrypt(text, shift); cout &lt;&lt; text;  delete[] text;</pre>	16 25 programmingisfun	oqnfqzllhmfhretm programmingisfun	oqnfqzllhmfhretm programmingisfun	✓

Your code failed one or more hidden tests.



## Câu hỏi 4

Đúng

Đạt điểm 1,00

**[Tiếng Việt]**

Viết hàm kiểm tra các phần tử trong mảng có duy nhất hay không

**Đầu vào:**

- int\* arr: mảng số tự nhiên
- int n: số lượng phần tử trong mảng

**Đầu ra:**

- bool: trả về true nếu các phần tử trong mảng là duy nhất, ngược lại trả về false

**Chú ý:** arr[i] nằm trong khoảng từ [0, 1000]

**[English]**

Write a function that determines if the elements in the given array is unique

**Input:**

- int\* arr: array of integer
- int n: the size of the array

**Output:**

- bool: return true if the elements in arr is unique, otherwise return false

**Note:** arr[i] is in the range of [0, 1000]

**For example:**



Test	Input	Result
<pre>int n; cin &gt;&gt; n; int* arr = new int[n]; for(int i = 0; i &lt; n; i++) {     cin &gt;&gt; arr[i]; } cout &lt;&lt; checkElementsUniqueness(arr, n); delete[] arr;</pre>	<pre>5 2 5 13 5 2</pre>	0
<pre>int n; cin &gt;&gt; n; int* arr = new int[n]; for(int i = 0; i &lt; n; i++) {     cin &gt;&gt; arr[i]; } cout &lt;&lt; checkElementsUniqueness(arr, n); delete[] arr;</pre>	<pre>3 17 10 25</pre>	1

Answer: (penalty regime: 0 %)

Reset answer

```
1 bool checkElementsUniqueness(int* arr, int n) {
2     // Kiểm tra từng phần tử với các phần tử sau nó
3     for (int i = 0; i < n - 1; ++i) {
4         for (int j = i + 1; j < n; ++j) {
5             if (arr[i] == arr[j]) {
6                 // Phần tử trùng nhau, không duy nhất
7                 return false;
8             }
9         }
10    }
11    // Tất cả các phần tử đều duy nhất
12    return true;
13 }
```







	Test	Input	Expected	Got	
✓	<pre>int n; cin &gt;&gt; n; int* arr = new int[n]; for(int i = 0; i &lt; n; i++) {     cin &gt;&gt; arr[i]; } cout &lt;&lt; checkElementsUniqueness(arr, n); delete[] arr;</pre>	<pre>5 2 5 13 5 2</pre>	0	0	✓
✓	<pre>int n; cin &gt;&gt; n; int* arr = new int[n]; for(int i = 0; i &lt; n; i++) {     cin &gt;&gt; arr[i]; } cout &lt;&lt; checkElementsUniqueness(arr, n); delete[] arr;</pre>	<pre>3 17 10 25</pre>	1	1	✓

Passed all tests! ✓



**Câu hỏi 5**

Không trả lời

Không chấm  
điểm**[Tiếng Việt ]**

Cho một số thập phân dương làm đầu vào, chúng ta cần triển khai hàm

```
long int decimalToBinary(int decimal_number){}
```

để chuyển đổi số thập phân dương đã cho thành số nhị phân tương đương.

Xin lưu ý rằng bạn không thể sử dụng từ khóa for, while, goto (ngay cả trong tên biến, comment).

Đối với bài tập này, chúng ta có #include <iostream> và sử dụng namespace std;

**[English ]**

Given a positive decimal number as input, we need to implement function

```
long int decimalToBinary(int decimal_number){}
```

to convert the given positive decimal number into equivalent binary number.

Please note that you can't using key work for, while, goto (even in variable names, comment).

For this exercise, we have #include <iostream> and using namespace std;

**For example:**



Test	Result
cout << decimalToBinary(20);	10100

Answer:

Reset answer

```
1 long int decimalToBinary(int decimal_number)
2 {
3     /*
4     * STUDENT ANSWER
5     */
6 }
```

