

| | |
|----------------------------|---|
| Đã bắt đầu vào lúc | Thứ sáu, 24 Tháng mười một 2023, 12:45 PM |
| Tình trạng | Đã hoàn thành |
| Hoàn thành vào lúc | Thứ sáu, 24 Tháng mười một 2023, 12:49 PM |
| Thời gian thực hiện | 3 phút 54 giây |
| Điểm | 4,00/4,00 |
| Điểm | 10,00 của 10,00 (100%) |

Câu hỏi 1

Chính xác

Điểm 1,00 của 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 |
|---|--------|
| const char* str = "abba"; cout << isPalindrome(str); | 1 |
| const char* str = "axz"; cout << isPalindrome(str); | 0 |

Answer: (penalty regime: 0 %)

Reset answer

```

1  bool isPalindrome(const char* str) {
2  int size = 0;
3  while (str[size] != '\0') size++;
4  char s[size];
5  for(int i = 0; i < size; i++){
6      s[i] = str[i];
7  }
8  char c[size];
9  int t = size;
10 for (int i = 0; i < size; i++){
11     c[i] = s[t-1];
12     t--;
13 }
14 for (int i = 0; i < size; i++){
15
16     if(c[i] != s[i]){
17         return 0;
18     }
19 }
20 return 1;
21 }

```


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

Passed all tests! ✓

Chính xác

Điểm cho bài nộp này: 1,00/1,00.

Câu hỏi 2

Chính xác

Điểm 1,00 của 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 >> n; cout << isSpecialNumber(n);</pre> | 23 | 1 |

| Test | Input | Result |
|---|-------|--------|
| <pre>int n; cin >> n; cout << isSpecialNumber(n);</pre> | 7 | 1 |

Answer: (penalty regime: 0 %)

Reset answer

```

1 bool prime(int n){
2     for(int i = 2; i*i<=n; i++){
3         if(n%i == 0){
4             return 0;
5         }
6     }
7     return (n>1);
8 }
9 bool isSpecialNumber(int n) {
10     int a = n;
11     if(prime(n) == 0){
12         return 0;
13     }
14     int sum = 0;
15     while(a>0){
16         int temp = a%10;
17         a = a/10;
18         sum = sum + temp;
19     }
20
21
22     if(prime(sum) == 0){
23         return 0;
24     }
25     return 1;
26 }
27

```


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

Passed all tests! ✓

Chính xác

Điểm cho bài nộp này: 1,00/1,00.

Câu hỏi 3

Chính xác

Điểm 1,00 của 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 >> n; int* arr = new int[n]; for(int i = 0; i < n; i++) { cin >> arr[i]; } cout << checkElementsUniqueness(arr, n); delete[] arr;</pre> | <pre>5 2 5 13 5 2</pre> | 0 |
| <pre>int n; cin >> n; int* arr = new int[n]; for(int i = 0; i < n; i++) { cin >> arr[i]; } cout << 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  int b[10005] = {0};
3  for(int i = 0; i < n; i++){
4      b[arr[i]]++;
5      if(b[arr[i]] > 1){
6          return 0;
7      }
8  }
9  return 1;
10 }
11
```


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

Passed all tests! ✓

Chính xác

Điểm cho bài nộp này: 1,00/1,00.

Câu hỏi 4

Chính xác

Điểm 1,00 của 1,00

[Tiếng Việt]

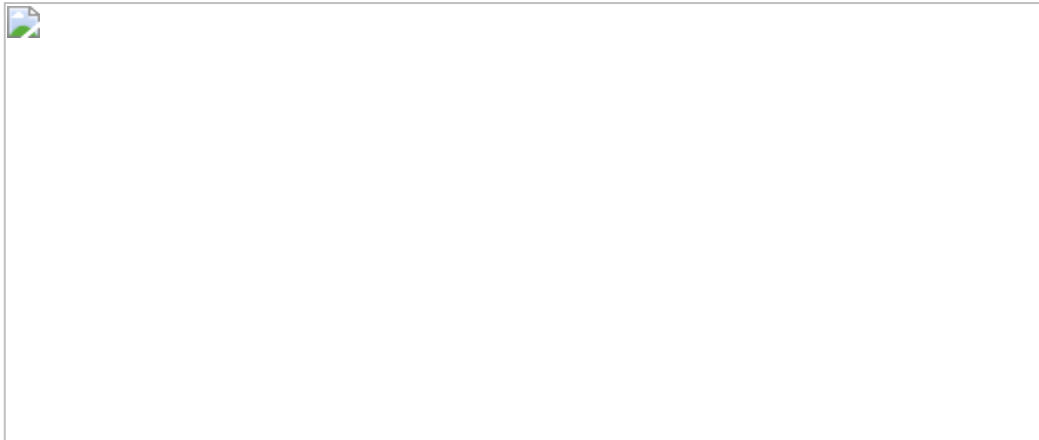
Viết hàm để mã hóa và giải mã một đoạn text theo phương pháp Caesar Cipher. Để mã hoá và giải mã một chuỗi ký tự text, ta cần một tham số có giá trị nguyên là shift.

Hàm mã hóa 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 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



[English]

Write a function to encrypt 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

For example:

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

| Test | Input | Result |
|---|-----------------------------------|--|
| <pre>int n, shift; cin >> n >> shift; char* text = new char[n+1]; for(int i = 0; i < n; i++) cin >> text[i]; text[n] = 0; encrypt(text, shift); cout << text << '\n'; decrypt(text, shift); cout << 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     while(shift >= 26){
3         shift = shift - 26;
4     }
5     int a;
6     for(int i = 0; text[i] != 0; i++){
7         int vietthuong = 0;
8         if(text[i] >= 'a' && text[i] <= 'z'){
9             vietthuong = 1;
10        }
11        // cout << text[i] << endl;
12        // cout << (int)text[i] << endl;
13        // text[i] = text[i] + shift;
14        int n = (int)text[i];
15        n = n + shift;
16        // cout << (int)text[i] << endl;
17        while(n > 'z' && vietthuong == 1){
18            n = n - 26;
19        }
20        while(n > 'Z' && vietthuong == 0){
```



```
21         n = n - 26;
22
23     }
24     text[i] = (char)n;
25     a = i;
26 }
27 text[a+2] = '\\0';
28 }
29
30 void decrypt(char* text, int shift) {
31     for(int i = 0; text[i] != 0; i++){
32         int vietthuong = 0;
33         if(text[i] >= 'a' && text[i] <= 'z'){
34             vietthuong = 1;
35         }
36         int n = (int)text[i];
37
38         n = n - shift;
39         while(n < 'a' && vietthuong == 1){
40             n = n + 26;
41         }
42         while(n < 'A' && vietthuong == 0){
43             n = n + 26;
44         }
45     }
46
47     text[i] = (char)n;
48 }
49
50 }
51
```

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

Passed all tests! ✓

Chính xác

Điểm cho bài nộp này: 1,00/1,00.

BÁCH KHOA E-LEARNING



WEBSITE

HCMUT

MyBK

BKSI

LIÊN HỆ

📍 268 Lý Thường Kiệt, P.14, Q.10, TP.HCM

☎ (028) 38 651 670 - (028) 38 647 256 (Ext: 5258, 5234)

✉ elearning@hcmut.edu.vn

Copyright 2007-2022 BKEL - Phát triển dựa trên Moodle