Bài tập ôn tập

1. Array

Mảng 1 chiều:

1) Xác định số lượng phần tử có giá trị chẵn
2) Xác định số lượng phần tử có giá trị lẻ
3) Xác định số lượng phần tử có giá trị dương
4) Xác định số lượng phần tử có giá trị âm
5) Xác định số lượng phần tử có giá trị là số chính phương
6) Xác định số lượng phần tử mà có giá trị là số nguyên tố
7) Xác định tổng các phần tử có giá trị chẵn
8) Xác định tổng các phần tử có giá trị lẻ
9) Xác định tổng các phần tử có giá trị dương
10) Xác định tổng các phần tử có giá trị âm
11) Xác định tổng các phần tử có giá trị là số nguyên tố
12) Xác định tổng các phần tử có giá trị là số chính phương
13) Xác định tổng của ba phần tử có giá trị chẵn lớn nhất
14) Xác định tổng của ba phần tử có giá trị chẵn nhỏ nhất
15) Xác định tổng của ba phần tử có giá trị lẻ lớn nhất
16) Xác định tổng của ba phần tử có giá trị lẻ nhỏ nhất
17) Xác định tổng của ba phần tử có giá trị âm lớn nhất
18) Xác định tổng của ba phần tử có giá trị âm nhỏ nhất
19) Xác định tổng của ba phần tử có giá trị dương lớn nhất
20) Xác định tổng của ba phần tử có giá trị dương nhỏ nhất

Mảng 2 chiều:

1) Đếm số lượng các phần tử dương nằm ở ma trận tam giác trên (hoặc dưới)
2) Tính tổng các phần tử nằm trên biên của ma trận
3) Xác định vị trí của phần tử đầu tiên (ưu tiên theo hàng và sau đó theo cột) vừa là phần tử lớn nhất trên dòng của nó đồng thời nhỏ nhất trên cột của nó
4) Đếm số lượng phần tử "yên ngựa" của ma trận, biết phần tử yên ngựa là phần tử lớn nhất trên dòng và nhỏ nhất trên cột tại vị trí đang xét (hoặc/và nhỏ nhất trên dòng và lớn nhất trên cột tại vị trí đang xét)
5) Đếm số lượng phần tử "hoàng hậu" của ma trận, biết phần tử hoàng hậu là phần tử lớn nhất trên dòng, trên cột và trên hai đường chéo đi qua nó.
6) Đếm số lượng phần tử "cực đại" trong ma trận, biết phần tử "cực trị" là phần tử lớn nhất so với các phần tử liền kề xung quanh nó.
7) Đếm số lượng phần tử "cực tiểu" trong ma trận, biết phần tử "cực tiểu" là phần tử nhỏ nhất so với các phần tử liền kề xung quanh nó.
8) Đếm số lượng phần tử "cực trị" trong ma trận, biết phần tử "cực trị" là phần tử lớn nhất hoặc nhỏ nhất so với các phần tử liền kề xung quanh nó.

9) Xác định vị trí của phần tử đầu tiên (ưu tiên theo hàng và sau đó theo cột) nằm ma trận ở tam giác trên mà là số chính phương và phần tử đối xứng với nó qua đường chéo chính cũng là số chính phương
10) Xác định vị trí của phần tử đầu tiên (ưu tiên theo hàng và sau đó theo cột) nằm ma trận ở tam giác trên mà là số nguyên tố và phần tử đối xứng với nó qua đường chéo chính cũng là số nguyên tố
11) Xoay ma trận 1 góc 90 độ
12) Xoay ma trận 1 góc 180 độ
13) Xoay ma trận 1 góc 270 độ
14) Đếm số lượng dòng giảm
15) Đếm số lượng dòng tăng
16) Đếm số lượng cột giảm
17) Đếm số lượng cột tăng
18) Tính tổng 2 ma trận
19) Tính hiệu 2 ma trận
20) Tính tích 2 ma trận

2. String

Xử lý ký tự:

1) Tìm kiếm 1 ký tự xem có tồn tại trong chuỗi không, nếu có thì trả về vị trí xuất hiện đầu tiên của ký tự đó.
2) Tìm kiếm 1 ký tự xem có tồn tại trong chuỗi không, nếu có thì trả về vị trí xuất hiện thứ k của ký tự đó.
3) Đếm số khoảng trắng trong chuỗi
4) Loại bỏ các khoảng trắng thừa trong chuỗi
5) c là một ký tự được nhập từ bàn phím. Tìm các chuỗi con (> 2 ký tự) chỉ chứa ký tự c và thay thế chuỗi các ký tự c lặp lại này thành chuỗi chỉ chứa 1 ký tự c duy nhất
6) Đếm số lần xuất hiện của một ký tự trong chuỗi
7) Đếm số lần xuất hiện lặp lại liên tục của một ký tự trong chuỗi
8) Đổi tất cả các ký tự trong chuỗi sang chữ thường (không dùng hàm strlwr)
9) Đổi tất cả các ký tự trong chuỗi sang chữ in hoa (không dùng hàm strupr)
10) Đổi chữ xem kẻ hoa và thường. Ví dụ: “ABCDefgH”” thành “abcdEFGh”
11) Kiểm tra xem chuỗi có chứa các ký tự số không.
12) Thay thế sự xuất hiện cuối cùng của ký tự c (nếu có tồn tại) bằng ký tự b trong chuỗi s
13) Đếm số nguyên âm trong chuỗi
14) Đếm số phụ âm trong chuỗi
15) Đếm số ký tự không thuộc bảng chữ cái
16) Đếm số ký tự là chữ số
17) Đếm số ký tự trong chuỗi dùng để mô tả giá trị của các số nguyên
18) Đếm số ký tự trong chuỗi dùng để mô tả giá trị của các số nguyên hoặc số thực
19) Đếm số ký tự trong chuỗi dùng để mô tả giá trị của các số nguyên hệ thập lục phân
20) Tính giá trị của một chuỗi các bit nhị phân
21) Tính giá trị của một chuỗi lưu trữ các bit hệ bát phân
22) Tính giá trị của một chuỗi lưu trữ các bit hệ thập lục phân

3. Function & Parameter passing

1. Kiểm tra xem một năm có là năm nhuận hay không?

   input: year:int

   output: true/false

   bool isLeapYear(int year)

2. Kiểm tra xem năm thuộc thế kỉ nào?

   input: year:int

   output: century:int

   void whichCentury(const int year, int& century)

3. Kiểm tra xem một số có phải số nguyên tố hay không?

   input: number: unsigned_int

   output: true/false

   bool isPrime(unsigned int* const pNum);

4. Một chuỗi được gọi là palindrom nếu chuỗi đó giống với chuỗi được đảo ngược từ chính nó. Ví dụ: “eye”, “noon”, “1991”... Kiểm tra xem một chuỗi có là palindrome hay không?

   input: str:string

   output: boolean

   bool isPalindrom(const char* str)

5. Tìm tích lớn nhất của hai phần tử liên tiếp trong một dãy? Ví dụ f([1,-3,4,7,3,2]) -> 4*7=28

   input: array or vector : [int

   output: largest_adjacent_product : int

   void (const int[] arr, int& largestAdjacentProduct)

6. Tính tổng hai ma trận cùng kích thước?

   input: [[int, [[int

   output: [[int

   void matrixProduct(const int arr1, const int** arr2, int* const resultArr)**

7. Kiểm tra xem liệu có thể chỉ bỏ đi nhiều nhất một phần tử trong mảng để thu được một mảng mới tăng chặt? [1,2,1,3,4] -> [1,2,3,4] bỏ đi số 1

   input: [int

   output: boolean

bool isIncrease(const int [] arr)

8. Tìm tất cả các chuỗi có số phần từ lớn nhất trong mảng các chuỗi? [“abc”, “a”, “ab”, “cds”] -> [“abc”, “cds”]

input: [string

output: [string

string* largestStrings(const string[] arr);

9. Đếm xem hai chuỗi có bao nhiêu ký tự giống nhau?f(“abcde”, “defs”) -> 2

input: [string, [string

output: int

int numOfSameChar(string a, string b);

10. Kiểm tra xem chuỗi này có là con chuỗi kia hay không? f(“abcd”, “bc”)->true

input: string, string

output: boolean

bool isSubstring(string& a, string& b);

11. Một mã số được coi là may mắn nếu tổng của một nửa các số đầu tiên bằng tổng của một nửa các số còn lại. Kiểm tra xem một mã số có may mắn hay không biết mã số có chẵn số chữ số?

input: string

output: boolean

void (string code, bool& isLucky)

12. Viết hàm chuyển số từ hệ thập phân sang nhị phân

input: int

output: char*

chat* (int num, int& lengthOfBinCode)

13. Kiểm tra chuỗi số có chứa số 0 hay không.

input: string

output: boolean

bool isContainsZero(string str)

14. Viết hàm xoay mảng sang trái k lần

input: [int, int

output: [int

void (const int [] arr, const int k, int*& const result);

15. Viết hàm xoay mảng sang phải k lần

    input: [int, int

    output: [int

    int* (const int [] arr, const int* k);

16. Viết hàm in ra các số nguyên tố nhỏ hơn n

    input: int

    output: void

    void (int num);

17. Kiểm tra một số nguyên n có phải là số chính phương hay không

    input: int

    output: boolean

    bool isSquareNumber(int num);

18. Liệt kê các ước số của số nguyên n

    input: int

    output: void

19. Tính tổng các số chẵn có trong đoạn a, b

    input: int, int

    output: int

20. Viết hàm in ra các số nguyên là bội của 3

    input: int

    output: void

4. Recursion

Thao tác trên mảng:

1) Tìm giá trị lớn nhất trong mảng
2) Tìm giá trị nhỏ nhất trong mảng
3) Tìm tổng của 1 mảng số nguyên
4) Tìm tổng các số chẵn của 1 mảng số nguyên
5) Tìm tổng các số lẻ của 1 mảng số nguyên
6) Kiểm tra xem mảng số nguyên chứa toàn số nguyên dương hay không?
7) Kiểm tra xem mảng số nguyên chứa toàn số nguyên âm hay không?
8) Kiểm tra xem mảng số nguyên chứa toàn số nguyên dương chẵn hay không?
9) Kiểm tra xem mảng số nguyên chứa toàn số nguyên âm lẻ hay không?
10) In các phần tử trong mảng

Thao tác trên chuỗi:

1) Tìm độ dài của chuỗi
2) Tìm kiếm 1 ký tự xem có tồn tại trong chuỗi không và trả về vị trí xuất hiện đầu tiên của ký tự đó.
3) In một chuỗi theo thứ tự ngược lại
4) Kiểm tra 1 chuỗi có phải là chuỗi nhị phân hay không
5) Chuyển 1 chuỗi số nhị phân sang số thập phân
6) Chuyển 1 chuỗi số thập lục phân sang số thập phân
7) Kiểm tra xem chuỗi có phải chuỗi palindrome hay không
8) Kiểm tra xem các ký tự khác nhau trong chuỗi đều có số lần xuất hiện là số chẵn hay không

5. Pointer

Singly linked list:

1. Đếm phần tử trong danh sách
2. Tính tổng hai phần tử lớn nhất trong danh sách
3. Thêm 1 node vào đầu danh sách liên kết đơn
4. Thêm 1 node vào cuối danh sách liên kết đơn
5. Xóa 1 node có 1 giá trị do người dùng chọn trong danh sách liên kết đơn
6. Xóa 1 node tại 1 index do người dùng chọn trong danh sách liên kết đơn
7. Xóa 1 node tại đầu danh sách liên kết đơn này và chuyển node đó sang đầu danh sách liên kết đơn thứ 2
8. Hoán đổi node đầu và node cuối của danh sách liên kết đơn với nhau
9. Hoán đổi 2 node bất kỳ trong danh sách liên kết đơn
10. Tìm node tại giữa danh sách liên kết đơn không biết trước kích thước chỉ với 1 vòng lặp
11. Ngắt danh sách liên kết đơn thành 2 phần, tại vị trí giữa của danh sách. Nếu chiều dài danh sách là lẻ, phần phía trước sẽ dài hơn phần sau một phần tử
12. Viết giải thuật sắp xếp 1 danh sách liên kết đơn theo thứ tự tăng dần giá trị
13. Xóa những phần tử trong danh sách liên kết đơn có giá trị lớn hơn 1 giá trị k
14. Xóa những phần tử trong danh sách liên kết đơn có giá trị lớn hơn giá trị tại phần tử có index do người dùng chọn
15. Cho một danh sách liên kết đơn đã được sắp thứ tự, hãy xóa những phần tử có giá trị trùng lặp với điều kiện chỉ được duyệt qua một lần
16. Cho hai danh sách liên kết đơn LList1 và LList2 , nối LList2 vào cuối của LList1
17. Cho hai danh sách liên kết đơn LList1 và LList2, mỗi danh sách đã được sắp xếp theo thứ tự tăng dần. Nối các node trong 2 danh sách đó lại thành một danh sách liên kết đơn duy nhất và vẫn giữ được thứ tự tăng dần của các node