BÀI TẬP MÔN CBASIC

Bài 1. Mảng, xâu, con trỏ

Mảng

Câu 1. Viết chương trình thực hiện việc đảo một mảng một chiều theo thứ tự ngược lại (lưu ý thực hiện đảo trên mảng, chứ không đơn thuần là duyệt theo thứ tự ngược lại)

Ví dụ : 1 2 3 4 5 7 9 10 đảo thành 10 9 7 5 4 3 2 1

Câu 2. Viết chương trình nhập vào một mảng số tự nhiên. Hãy in ra màn hình:

- Dòng 1 : gồm các số lẻ, tổng cộng có bao nhiêu số lẻ.
- Dòng 2 : gồm các số chẵn, tổng cộng có bao nhiêu số chẵn.

Câu 3. Viết chương trình nhập từ bàn phím hai số nguyên n và m. Sau đó nhập n số nguyên lưu vào mảng a, nhập m số nguyên lưu vào mảng b

- Hiển thị ra màn hình số lượng các phần tử chung của hai mảng này, và danh sách các phần tử đó
- Sắp xếp mảng a và b theo thứ tự tăng dần. Hiển thị kết quả ra màn hình
- Từ hai mảng a và b, gộp lại thành mảng c sao cho các phần tử trong c theo thứ tự tăng dần. Hiển thị mảng c ra màn hình

Xâu kí tự

Câu 4. Viết chương trình nhập một xâu ký tự và kiểm tra xem chuỗi đó có đối xứng không. Ví dụ : Xâu ABCDEDCBA là xâu đối xứng.

Câu 5. Nhập vào một xâu bất kỳ, hãy đếm số lần xuất hiện của mỗi chữ cái (từ a-z)

Câu 6. Viết chương trình nhập vào một xâu.

- In ra màn hình từ bên trái nhất và phần còn lại của xâu. Ví dụ: “Nguyễn Văn Long” in ra thành:

Nguyễn

Văn Long

Câu 7. Viết chương trình nhập vào một xâu rồi xuất xâu đó ra màn hình dưới dạng mỗi từ một dòng.

Ví dụ: “Nguyễn Văn Long”

In ra :

Nguyễn

Văn

Long

Con trỏ

Câu 8. Nhập, thực hiện và quan sát kết quả của chương trình sau

```
#include<stdio.h>
#include <conio.h>
int main()
  {
    int A = 1;
    int B = 2;
    int C = 3;
    int *P1, *P2;
    P1=&A;
    P2=&C;
    *P1=(*P2)++;
    printf("%d %d",*P1, *P2);
    P1=P2;
    P2=&B;
    *P1-=*P2;
    printf("%d %d",*P1, *P2);
    ++*P2;
    *P1*=*P2;
    printf("%d %d",*P1, *P2);
    A=++*P2**P1;
    P1=&A;
    printf("%d",A);
    *P2=*P1/=*P2;
```

```
    printf("%d %d",*P1, *P2);
    return 0;
}
```

Câu 9. Cho p là con trỏ trỏ tới mảng A:

```
int A[] = {12, 23, 34, 45, 56, 67, 78, 89, 90};
int *P;
P = A;
```

Cho biết giá trị của các biểu thức sau:

a. *P+2
b. *(P+2)
c. &P+1
d. &A[4]-3
e. A+3
f. &A[7]-P
g. P+(*P-10)
h. *(P+*(P+8)-A[7])

Câu 10. Viết chương trình đọc vào một số nguyên x và mảng nguyên a, sau đó loại bỏ tất cả các phần tử bằng x trong mảng. Dùng hai con trỏ p1 và p2 để duyệt mảng.

.

Bài 2. File

Câu 11. Tạo một file văn bản lab1.txt với nội dung tùy ý. Sau đó, viết chương trình đọc lần lượt từng kí tự trong file trên và ghi vào file mới có tên lab2.txt

Câu 12. Viết chương trình nhập từ bàn phím N số thực lưu vào một mảng (N < 100 và N được nhập từ bàn phím). Sau đó ghi ra một file văn bản có địa chỉ là "float.txt" theo quy cách: dòng đầu tiên lưu số lượng các số thực, các dòng tiếp theo lưu các số thực, mỗi số lưu trên một dòng. Đọc lại tệp văn bản đó và lưu các số thực lớn hơn 5 vào tệp "float2.txt" theo quy cách giống như tệp "float.txt". Lưu mã nguồn chương trình với tên file_1.C.

Câu 13. Viết chương trình sao chép nội dung tệp mã nguồn chương trình C có tên là file_1.C sang tệp có tên là file_2.C bằng 2 cách (cách 1 sử dụng fread, fwrite; cách 2 là các hàm khác như fprintf, fscanf…)

Câu 14. Một từ điển "Anh-Việt" đơn giản có dữ liệu được lưu trữ trong file "data.txt" với định dạng như sau: từ tiếng Anh rồi đến dấu tab sau đó đến từ tiếng Việt

Ví dụ: school truong hoc (khoảng ở giữa tương ứng với 1 dấu tab)

Giả sử rằng trong file này, các từ tiếng Anh không có dấu cách, các từ tiếng Việt có thể có dấu cách, độ dài tối đa của từ tiếng Anh và tiếng Việt là 20 kí tự. Hãy tạo file "data.txt" với ít nhất 5 từ tiếng Anh. Sau đó đọc nội dung của các từ tiếng Anh lưu vào một mảng, các từ tiếng Việt lưu vào một mảng. Hiển thị 2 mảng này ra màn hình.

Câu 15. Thông tin về một thí sinh bao gồm: số báo danh (tối đa 10 kí tự), họ và tên (tối đa 30 kí tự), điểm thi (số thập phân). Hãy viết chương trình nhập từ bàn phím thông tin các thí sinh cho đến khi nhập vào số bao danh là -1. Thông tin mỗi thí sinh được ghi ra file văn bản "thisinh.txt" (không sử dụng mảng để lưu ở bước này). Sau đó mở lại file để đọc thông tin vào mảng ts. Hiển thị mảng ts ra màn hình. Tiếp đó duyệt mảng ts để lưu thông tin những thí sinh có điểm thi >=5 vào file văn bản "thisinh2.txt"

Câu 16. Nội dung & yêu cầu như bài trên, chỉ thay file văn bản "thisinh.txt" và "thisinh2.txt" bằng 2 file nhị phân "thisinh.dat" và "thisinh2.dat"

Câu 17. Sửa lại bài 12. Sau khi nhập từ bàn phím số nguyên N và N số thực. Hãy ghi mảng gồm N số thực đó vào một file nhị phân có tên "sothuc.dat". tiếp theo, đọc lại file nhị phân này và ghi các số thực lớn hơn 5 vào file nhị phân có tên "sothuc2.dat"

Bài 3. Danh sách liên kết

Câu 18. Thông tin về một sinh viên gồm có: **Mã số SV** là xâu tối đa 10 kí tự, **Họ tên** là xâu tối đa 30 kí tự, **điểm thi** Cbasic là số thực.
Viết chương trình thực hiện công việc sau:

- Định nghĩa cấu trúc Sinh viên
- Nhập một số nguyên dương n thỏa mãn 2<= n <=10 từ bàn phím. Sau đó nhập lần lượt thông tin của n sinh viên và lưu vào một danh sách liên kết.
- Sau khi nhập xong hiển thị thông tin các SV này ra màn hình

Câu 19. Định nghĩa cấu trúc sinh viên như câu hỏi trên. Sau đó xây dựng chương trình có giao diện menu thực hiện công việc sau

- Khi ấn 1 cho phép nhập từ bàn phím thông tin của 1 sinh viên, sau đó chèn vào đầu danh sách
- Khi ấn 2 cho phép nhập từ bàn phím mã số SV cần tìm kiếm. Nếu có, hiển thị tất cả thông tin về SV, ngược lại thông báo không tìm thấy.
- Khi ấn 3 cho phép nhập từ bàn phím mã số SV cần xóa. Nếu không có SV này thì thông báo xóa không thành công vì không tồn tại SV như vậy
- Khi ấn 4 cho phép hiển thị tất cả thông tin SV trong danh sách
- Khi ấn 5 cho phép thoát khỏi chương trình

Câu 20. Chỉnh sửa bài trên ở menu lựa chọn 1 như sau

- Khi ấn 1 cho phép nhập từ bàn phím thông tin của 1 sinh viên, sau đó chèn vào danh sách sao cho điểm của SV theo thứ tự tăng dần (từ đầu đến cuối danh sách)

Câu 21. Chỉnh sửa bài trên với các chức năng bổ sung sau

- Khi ấn 6 cho phép ghi toàn bộ nội dung thông tin các sinh viên ra một file nhị phân có tên “sinhvien.dat”
- Khi ấn 7 thực hiện đọc thông tin file nhị phân “sinhvien.dat” vào một mảng cấu trúc. Hiển thị thông tin các sinh viên trong mảng có điểm thi đạt (lớn hơn hoặc bằng 4) và tỷ lệ số SV thi đạt.

Câu 22. Nhập từ bàn phím hai số nguyên n, m. Sau đó nhập n số nguyên lưu vào danh sách liên kết thứ nhất, nhập m số nguyên lưu vào danh sách liên kết thứ hai. Thực hiện gộp các phần tử của hai danh sách này thành một danh sách sau cho các phần tử trong danh sách thu được có giá trị tăng dần

Tuần 4+5. Stack, Queue

Câu 23. Đọc nội dung của tệp "sinhvien.dat" trong câu 20 và lưu vào một stack cài đặt bằng mảng. Sau đó đọc lần lượt các phần tử của stack và ghi vào file "sinhvien2.dat". Như vậy thông tin sinh viên trong file sinhvien.dat sẽ được ghi theo thứ tự ngược lại trong file sinhvien2.dat

Câu 24. Thực hiện như câu 22 nhưng stack cài đặt bằng danh sách liên kết

Câu 25. Thực hiện như câu 22 nhưng lưu vào queue cài đặt bằng mảng.

Câu 26. Thực hiện như câu 22 nhưng lưu vào queue cài đặt bằng danh sách liên kết

Câu 27. Palindromes là các xâu ký tự mà đọc nó từ trái sang phải thì cũng giống như khi đọc từ phải sang trái. Ví dụ: noon, madam, radar. Viết chương trình thực hiện kiểm tra xem một xâu có phải là xâu đối xứng (palindromes) không theo cách sau:

Đầu vào: nhập một xâu ký tự đầu vào

Đầu ra: Xác định xâu ký tự là đối xứng hay không

Giải quyết:

A. Đọc các ký tự trong xâu rồi lưu đồng thời vào một Stack và một Queue

B. Lấy lần lượt các ký tự đồng thời từ Stack , Queue và so sánh

   i. Nếu có một cặp ký tự được lấy ra không giống nhau → Xâu đầu vào không phải palindromes

   ii. Nếu tất cả các cặp ký tự lấy ra đều giống nhau → Xâu đầu vào là palindromes

Câu 28. Kiểm tra tính cân đối của các ký hiệu đóng mở trong phân tích mã nguồn. Khi viết chương trình bằng các ngôn ngữ như C hay Java, một yêu cầu cơ bản để chương trình không mắc lỗi cú pháp là các cặp ngoặc mở và đóng phải đối xứng và tương ứng với nhau theo đúng thứ tự ví dụ như sau {…. (……[…..]….)… }. Với đoạn mã chứa {…. […. (…]….)… } chương trình dịch chắc chắn sẽ báo lỗi. Hãy viết một chương trình (giao diện menu) có các chức năng sau (1 điểm cho giao diện)

- Nhập tên và đường dẫn file mã nguồn sau đó hiển thị nội dung của mã nguồn ra màn hình. (2 điểm)
- Kiểm tra tính đúng đắn về cú pháp của các ký tự ngoặc (nhọn, vuông, tròn). Nếu sai hãy in ra thông báo trong mọi trường hợp: (3 – 3.5 điểm)
- Lỗi thiếu ngoặc đóng và là dạng ngoặc nào (Chỉ có ngoặc mở mà không có ngoặc đóng)

- Lỗi ngoặc đóng không tương ứng với ngoặc mở gần nhất – cùng ký tự ngoặc gây lỗi.
- In ra màn hình chương trình nguồn và đánh dấu tại vị trí gây lỗi. (*Chức năng cộng điểm:* In ra được vị trí dòng, cột của ký tự gây lỗi.)

Gợi ý về thuật toán: Sử dụng cấu trúc dữ liệu ngăn xếp cho phép giải quyết bài toán này một cách đơn giản. Nếu coi mã nguồn như một mảng ký tự (không quá 500000 ký tự) thì:

- Đọc lần lượt các ký tự, nếu là ký tự ngoặc mở thì PUSH vào ngăn xếp
- Nếu gặp ký tự ngoặc đóng thì POP một ký tự ra khỏi ngăn xếp, đối sánh chúng xem có đúng là cặp ký tự đóng mở tương ứng hay không.
- Khi đi hết mã nguồn mà ngăn xếp vẫn còn ký tự có nghĩa chương trình mắc lỗi.

Tuần 6+7. Tìm kiếm

Câu 29. Viết chương trình cho phép nhập vào một mảng các số nguyên từ bàn phím. Sau đó nhập vào một số nguyên X để tìm kiếm. Hãy hiển thị ra chỉ số tất cả các phần tử trong mảng có giá trị X. Nếu không thấy in ra thông báo không tìm được. Sử dụng giải thuật tìm kiếm tuần tự.

Câu 30. Viết chương trình tìm kiếm gần đúng như sau. Tạo một file văn bản trong đó mỗi dòng là một xâu có độ dài <=30 kí tự. Thực hiện nhập từ bàn phím một từ cần tìm. Hiển thị ra màn hình các xâu trong file chứa từ này

Ví dụ: từ nhập vào là computer. Các xâu thỏa mãn như: computer, computers, super computer…

Câu 31. Sử dụng cấu trúc Sinh viên ở câu 17. Viết chương trình giao diện menu có các chức năng sau (1đ cho giao diện)

- Nhập thông tin các sinh viên từ bàn phím và lưu vào file "SV.dat" (1 đ). Khi nhập có kiểm tra xem điểm có thuộc phạm vi từ 0 đến 10 không, nếu không yêu cầu nhập lại điểm cho đến khi thỏa mãn (0.5đ). Quá trình nhập sẽ dừng khi nhập họ tên là xâu "###". (0.5 đ). Lưu ý không dùng mảng để lưu trữ ở bước này.
- Đọc dữ liệu từ file "SV.dat" ra một mảng (1đ)
- Sắp xếp mảng theo thứ tự tăng dần của maSV. Sau đó hiển thị kết quả ra màn hình (1đ)
- Nhập vào một **maSV** cần tìm. Sử dụng giải thuật tìm kiếm nhị phân, cho biết có SV này trong mảng không. Nếu có hiển thị ra họ tên, điểm (1đ)

Câu 32. Để quản lý thông tin về các sản phẩm, người ta lưu trữ vào file các bản ghi sản phẩm gồm mã sản phẩm (maSP – kiểu int, có giá trị trong khoảng từ 101 đến 500), tên sản phẩm (tenSP – kiểu xâu), mô tả sản phẩm (motaSP – kiểu xâu), giá (giaSP – kiểu int), số lượng (SL – kiểu int). Viết chương trình (giao diện menu) có các chức năng sau (1 điểm cho giao diện)

- Không dùng mảng, hãy nhập thông tin về các sản phẩm theo thứ tự tăng dần của mã sản phẩm và ghi vào file "SANPHAM.TXT" (1đ). Quá trình nhập liệu có kiểm tra mã sản phẩm trong khoảng từ 101 đến 500 (0.5đ). Quá trình nhập liệu kết thúc khi mã sản phẩm không phải là số (0.5đ)
- Đọc dữ liệu từ file "SANPHAM.TXT" ra 1 mảng (1đ)
- Tìm bản ghi với maSP nhập từ bàn phím và in ra màn hình bản ghi đó. Hãy viết thuật toán tìm kiếm hiệu quả nhất để thực hiện công việc này (1đ). In ra màn hình số phép so sánh cần thực hiện để tìm được bản ghi này (0.5đ)

Tuần 8. Cây

Câu 33. Tính Tần suất xuất hiện của các từ trong văn bản.

Viết chương trình wcount có tính năng sau:

1. Nhân tham số từ dòng lệnh là một file văn bản. In nội dung file. (1.5 điểm)
2. Yêu cầu người dùng nhập vào một từ, in ra tần suất xuất hiện của từ đó trong văn bản nếu có. Nếu không cũng in ra thông báo. (1 điểm)
3. In ra thống kê (dạng bảng) lần lượt các từ có trong văn bản theo thứ tự từ điển và số lần xuất hiện của chúng trong văn bản. Các từ được quy ước là các dãy ký tự (không chứa ký tự đặc biệt) cách nhau với dấu space, xuống dòng, tab. (2.5 điểm)

Yêu cầu về cấu trúc dữ liệu:

Phải dùng cây nhị phân tìm kiếm để lưu các từ cùng tần suất xuất hiện của chúng. Chương trình kết thúc thì bộ nhớ động sử dụng cũng phải được giải phóng.

Gợi ý: Cách đơn giản để đọc lần lượt các từ trong file văn bản là liên tục dùng hàm fscanf thay vì viết hàm để nhận dạng từ.

```
char buf[MAXLENGTH];
.....
fscanf(f,"%s",buf);
```

Câu 34. Một từ điển dịch "Anh-Việt" đơn giản có dữ liệu được lưu trữ trong file "data.txt" với định dạng như sau: từ tiếng Anh rồi đến dấu tab sau đó đến từ tiếng Việt

Ví dụ: school truong hoc (khoảng ở giữa tương ứng với 1 dấu tab)

Giả sử rằng các từ tiếng Anh đều là các từ không có chứa dấu cách và không có sự trùng lặp trong file từ điển. Từ tiếng Việt có thể có chứa dấu cách. Các từ tiếng Anh, tiếng Việt ở dạng chữ thường và có độ dài tối đa là 20 kí tự.

Tạo cấu trúc để lưu trữ thông tin về từ điển và xây dựng chương trình có ***giao diện menu*** thực hiện các công việc sau:

1. Đọc file data.txt và lưu vào một cây nhị phân tìm kiếm. Việc so sánh được thực hiện dựa trên từ tiếng Anh.
2. Hiển thị toàn bộ dữ liệu trên cây theo thứ tự tăng dần của từ tiếng Anh.

3. Dịch Anh-Việt. Nhập vào một câu chỉ gồm các từ tiếng Anh ở dạng viết hoa hoặc viết thường các kí tự, chương trình dịch ra nghĩa tiếng Việt tương ứng.

- Đầu tiên thực hiện tách câu ra thành các từ đơn, chuyển từ về dạng chữ thường và hiển thị các từ này ra màn hình, mỗi từ trên một dòng.
- Tiếp đó thực hiện dịch câu. Có các trường hợp xảy ra. Trường hợp 1 nếu tồn tại một từ không thấy trên cây thì thì hiển thị "***Thiếu từ***" và cho phép người sử dụng bổ sung từ này vào cây, việc bổ sung vẫn đảm bảo cây thu được là cây nhị phân tìm kiếm. Trường hợp còn lại sẽ hiển thị ra nghĩa tiếng Việt.

Ví dụ nếu câu đầu vào là "**I like PHone**" thì dịch ra là "**toi thich dien thoai**"

Sau khi kết thúc chương trình ghi lại cây vào từ điển để cập nhật những bổ sung (nếu có)

Gợi ý. Để đọc file data.txt. Với mỗi dòng, sinh viên có thể dùng hàm fscanf để đọc từ tiếng Anh, rồi dùng fgetc để đọc dấu tab, sau đó dùng hàm fgets để đọc từ tiếng Việt còn lại trong dòng.

Câu 35. Xây dựng chương trình Kiểm tra đăng nhập và quản trị mật khẩu với các yêu cầu chức năng sau:

- Khi bắt đầu, chương trình hỏi người sử dụng nhập vào username và password. Nếu là người dùng thông thường, khi mật khẩu đúng thì chương trình in ra thông báo đăng nhập thành công và người dùng có thể thay đổi lại mật khẩu của mình. Nếu là người quản trị (username = admin) thì khi đăng nhập thành công, anh ta có thể chạy các chức năng sau:
    - Thêm một người dùng vào hệ thống (nhập từ bàn phím tên và mật khẩu)
    - Đổi mật khẩu cho một tài khoản (trong trường hợp người dùng quên mật khẩu cũ)
    - In ra danh sách người dùng (theo username)
- Trong trường hợp đăng nhập không thành công thì chương trình báo lỗi và yêu cầu nhập lại, nếu quá ba lần nhập sai chương trình sẽ tự động thoát.
- Yêu cầu về mật khẩu (password): dài ít nhất 6 ký tự, chỉ gồm chữ cái và chữ số cùng các ký tự _ (gạch dưới), $ (đô la). Yêu cầu về username: không chứa dấu cách.

- Chương trình phải lưu thông tin đăng nhập vào tập tin trước khi thoát ra và khi chạy lại các tài khoản mới sẽ có tác dụng

Yêu cầu về cấu trúc dữ liệu và cấu trúc file:

- Chương trình phải sử dụng cấu trúc cây nhị phân tìm kiếm để tìm kiếm người dùng và đối sánh mật khẩu, cũng như thêm người dùng mới vào hệ thống. Sinh viên được quyền sử dụng lại (có sửa đổi) thư viện hàm xử lý cây nhị phân tìm kiếm đã viết.
- File lưu thông tin đăng nhập (pass.txt) đơn giản là file văn bản, mỗi dòng văn bản chứa: Username password. Ví dụ

  dangnt bi456

  trungbt $mimosa929