

BÀI TẬP LỚN

CS1-255

Các nhóm sinh viên chọn một trong các đề tài dưới đây, từ đó xây dựng một ứng dụng trên nền console thực hiện các yêu cầu đề bài đưa ra. Sinh viên cần phân tích kỹ lưỡng yêu cầu đề bài, tự đưa ra giải pháp phù hợp.

Tiêu chí đánh giá bài tập lớn:

- Chương trình chạy đúng, có đủ các chức năng theo yêu cầu
- Mã nguồn chương trình được trình bày đẹp. Các chú thích đầy đủ, rõ ràng.
- Chương trình có khả năng xử lý các trường hợp dữ liệu đưa vào chưa được chuẩn hóa hay có lỗi tồn tại.
- Đã tiến hành các kiểm thử để bảo đảm chương trình chạy ổn định.
- Phân chia công việc trong nhóm hợp lý.
- Nộp bài tập lớn: Nộp lên Teams – Assignment MiniProject: báo cáo phân tích và cài đặt + source code (.zip)

Số đề tài được gán cho mỗi nhóm = STT của nhóm / 5 lấy dư + 1.

DANH SÁCH ĐỀ TÀI

Đề tài 1- CHUYỂN HỆ CƠ SỐ DÀNH CHO SỐ NGUYÊN

Xây dựng một chương trình cho phép người dùng nhập vào từ bàn phím một dãy kí tự thuộc một hệ cơ số cho trước. Người dùng được phép chọn hệ cơ số mình mong muốn: 2 – 8 – 10 – 16.

Xây dựng menu lựa chọn, 0 – Exit.

Các yêu cầu:

- Chương trình cho phép người dùng nhập hệ cơ số input, hệ cơ số output, nhập vào chuỗi kí tự biểu diễn một số tương ứng trong hệ cơ số input, số byte sẽ dùng để biểu diễn cho số đó. Ví dụ người dùng nhập 2 cho hệ nhị phân, nhập chuỗi nhị phân input: 11010, nhập 10 cho số byte sẽ dùng để biểu diễn hệ cơ số input và output.
- Chương trình sẽ hiển thị biểu diễn của số input với số bit được nhập ở trên (1byte = 8 bit), các bit trống sẽ điền 0 ở trước.
- Chương trình output ra số output tương ứng trong hệ cơ số tương ứng của số ban đầu (input).
- Chú ý, với các số có dấu, chương trình sẽ dành 1 bit đầu tiên biểu diễn dấu, 0: số dương và 1: số âm.
- Chương trình cần có cơ chế check lỗi input, và xử lý thông báo khi tràn số.

Các số trong hệ cơ số 16 thì 10: A, 11: B, 12: C, 13: D, 14: E và 15: F.

Đề tài 2: BIỂU THỨC TIỀN TỔ

Biểu thức tiền tố là một cú pháp trong biểu diễn toán học trong đó các toán tử (operators) được chuyển lên trước các toán hạng (operands).

Ví dụ: dạng infix của biểu diễn toán học một biểu thức như sau: $(a+b) * (c+d)$.

Dạng tiền tố sẽ chuyển các toán tử lên trước các toán hạng, ví dụ cho biểu thức $a+b \rightarrow$ prefix sẽ là: $+ a b$

Tương tự, $c + d \rightarrow$ prefix là $+ c d$.

Prefix của biểu thức ban đầu: $* + a b + c d$

Xây dựng chương trình thực hiện các yêu cầu sau đây:

+ Cho phép người dùng nhập vào một chuỗi các kí tự biểu diễn biểu thức thông thường (dạng infix), các toán hạng sẽ là các số cụ thể (giới hạn là các số nguyên, được phép là số âm). Ví dụ: $(12 + 10) * (17 - 8)$.

Chương trình sẽ phải đọc và check đúng dạng của biểu thức, dạng như sau sẽ thông báo lỗi cú pháp: $(12\ 10) - 7$

+ Cho phép chuyển đổi biểu thức infix sang dạng prefix tương ứng

+ Tính giá trị của biểu thức prefix.

Xây dựng menu:

1. Nhập biểu thức infix
2. Tính giá trị của biểu thức prefix
3. Exit.

Khi người dùng lựa chọn 1, chương trình sẽ check biểu thức infix được nhập từ bàn phím, thông báo lỗi cú pháp nếu có và hiển thị lại menu cho người dùng. Nếu không có lỗi thì chương trình sẽ hiển thị biểu thức prefix tương ứng, menu để người dùng chọn.

Khi người dùng lựa chọn 2, chương trình sẽ tính toán giá trị của biểu thức infix, nếu biểu thức infix chưa được nhập, thì sẽ hiển thị thông báo yêu cầu nhập biểu thức infix và hiển thị menu để người dùng tiếp tục thao tác.

Khi người dùng lựa chọn 3, thoát khỏi chương trình.

Đề tài 3: BIỂU THỨC HẬU TỔ

Biểu thức hậu tố là một cú pháp trong biểu diễn toán học trong đó các toán tử (operators) được chuyển xuống dưới các toán hạng (operands).

Ví dụ: dạng infix của biểu diễn toán học một biểu thức như sau: $(a+b) * (c+d)$.

Dạng hậu tố sẽ chuyển các toán tử ra phía sau các toán hạng, ví dụ cho biểu thức $a+b \rightarrow$ postfix sẽ là: $a\ b\ +$

Tương tự, $c\ +\ d \rightarrow$ postfix là $c\ d\ +$.

Postfix của biểu thức ban đầu: $a\ b\ +\ c\ d\ +\ *$

Xây dựng chương trình thực hiện các yêu cầu sau đây:

+ Cho phép người dùng nhập vào một chuỗi các kí tự biểu diễn biểu thức thông thường (dạng infix), các toán hạng sẽ là các số cụ thể (giới hạn là các số nguyên, được phép là số âm). Ví dụ: $(12 + 10) * (17 - 8)$.

Chương trình sẽ phải đọc và check đúng dạng của biểu thức, dạng như sau sẽ thông báo lỗi cú pháp: $(12\ 10) - 7$

+ Cho phép chuyển đổi biểu thức infix sang dạng postfix tương ứng

+ Tính giá trị của biểu thức postfix.

Xây dựng menu:

1. Nhập biểu thức infix
2. Tính giá trị của biểu thức postfix
3. Exit.

Khi người dùng lựa chọn 1, chương trình sẽ check biểu thức infix được nhập từ bàn phím, thông báo lỗi cú pháp nếu có và hiển thị lại menu cho người dùng. Nếu không có lỗi thì chương trình sẽ hiển thị biểu thức postfix tương ứng, menu để người dùng chọn.

Khi người dùng lựa chọn 2, chương trình sẽ tính toán giá trị của biểu thức infix, nếu biểu thức infix chưa được nhập, thì sẽ hiển thị thông báo yêu cầu nhập biểu thức infix và hiển thị menu để người dùng tiếp tục thao tác.

Khi người dùng lựa chọn 3, thoát khỏi chương trình.

Đề tài 4: Chuyển đổi từ biểu thức tiền tố (prefix) sang biểu thức thông thường (infix)

Biểu thức tiền tố là một cú pháp trong biểu diễn toán học trong đó các toán tử (operators) được chuyển lên trước các toán hạng (operands).

Ví dụ: dạng infix của biểu diễn toán học một biểu thức như sau: $(a+b) * (c+d)$.

Dạng tiền tố sẽ chuyển các toán tử lên trước các toán hạng, ví dụ cho biểu thức $a+b \rightarrow$ prefix sẽ là: $+ a b$

Tương tự, $c + d \rightarrow$ prefix là $+ c d$.

Prefix của biểu thức ban đầu: $* + a b + c d$

Xây dựng chương trình thực hiện các yêu cầu sau đây:

+ Cho phép người dùng nhập vào một chuỗi các kí tự biểu diễn biểu thức tiền tố (prefix) các toán hạng sẽ là các số cụ thể (giới hạn là các số nguyên, được phép là số âm). Ví dụ: $* 12 - 13$

Chương trình sẽ phải đọc và check đúng dạng của biểu thức, dạng như sau sẽ thông báo lỗi cú pháp.

+ Cho phép chuyển đổi biểu thức prefix sang dạng infix tương ứng

+ Tính giá trị của biểu thức infix.

Xây dựng menu:

1. Nhập biểu thức prefix
2. Tính giá trị của biểu thức infix
3. Exit.

Khi người dùng lựa chọn 1, chương trình sẽ check biểu thức prefix được nhập từ bàn phím, thông báo lỗi cú pháp nếu có và hiển thị lại menu cho người dùng. Nếu không có lỗi thì chương trình sẽ hiển thị biểu thức infix tương ứng, menu để người dùng chọn.

Khi người dùng lựa chọn 2, chương trình sẽ tính toán giá trị của biểu thức infix, nếu biểu thức prefix chưa được nhập, thì sẽ hiển thị thông báo yêu cầu nhập biểu thức prefix và hiển thị menu để người dùng tiếp tục thao tác.

Khi người dùng lựa chọn 3, thoát khỏi chương trình.

Đề tài 5: Chuyển đổi từ biểu thức hậu tố (postfix) sang biểu thức thông thường (infix)

Biểu thức hậu tố là một cú pháp trong biểu diễn toán học trong đó các toán tử (operators) được chuyển xuống dưới các toán hạng (operands).

Ví dụ: dạng infix của biểu diễn toán học một biểu thức như sau: $(a+b) * (c+d)$.

Dạng hậu tố sẽ chuyển các toán tử ra phía sau các toán hạng, ví dụ cho biểu thức $a+b \rightarrow$ postfix sẽ là: $a b +$

Tương tự, $c + d \rightarrow$ postfix là $c d +$.

Postfix của biểu thức ban đầu: $a b + c d + *$

Xây dựng chương trình thực hiện các yêu cầu sau đây:

+ Cho phép người dùng nhập vào một chuỗi các kí tự biểu diễn biểu thức hậu tố (dạng postfix), các toán hạng sẽ là các số cụ thể (giới hạn là các số nguyên, được phép là số âm).

Chương trình sẽ phải đọc và check đúng dạng của biểu thức, dạng như sau sẽ thông báo lỗi cú pháp nếu có.

+ Cho phép chuyển đổi biểu thức postfix sang dạng infix tương ứng

+ Tính giá trị của biểu thức infix.

Xây dựng menu:

1. Nhập biểu thức postfix
2. Tính giá trị của biểu thức infix
3. Exit.

Khi người dùng lựa chọn 1, chương trình sẽ check biểu thức postfix được nhập từ bàn phím, thông báo lỗi cú pháp nếu có và hiển thị lại menu cho người dùng. Nếu không có lỗi thì chương trình sẽ hiển thị biểu thức infix tương ứng, menu để người dùng chọn.

Khi người dùng lựa chọn 2, chương trình sẽ tính toán giá trị của biểu thức infix, nếu biểu thức postfix chưa được nhập, thì sẽ hiển thị thông báo yêu cầu nhập biểu thức postfix và hiển thị menu để người dùng tiếp tục thao tác.

Khi người dùng lựa chọn 3, thoát khỏi chương trình.