

<b>12. atol(const char *s)</b> Hàm đổi chuỗi s có chứa ký số ở vị trí đầu chuỗi thành một số nguyên. Hàm trả về kết quả là 1 số nguyên kiểu long. Nếu chuỗi s không chứa ký số hoặc không nằm ở vị trí đầu chuỗi, hàm trả về 0.	<pre>char s[] = "8055355"; long i = atol(s1);</pre> <b>Kết quả:</b> i = 8055355
<b>13. atof(const char *s)</b> Hàm đổi chuỗi s có chứa ký số ở vị trí đầu chuỗi thành một số thực. Hàm trả về kết quả kiểu float. Nếu chuỗi s không chứa ký số hoặc không nằm ở vị trí đầu chuỗi, hàm trả về 0.0	<pre>char s[] = "1234.03"; float i = atof(s);</pre> <b>Kết quả:</b> i = 1234.03



## Bài tập thực hành

### Ghi nhớ:

1. Định dạng ký tự được đặt trong cặp nháy đơn "và có đặc tả là %c.
2. Định dạng chuỗi ký tự được đặt trong cặp nháy kép "" và có đặc tả là %s.
3. Lệnh scanf không thể đọc khoảng trắng trong chuỗi, hãy thay thế bằng lệnh gets.
4. Lệnh printf không thể tự động xuống dòng, hãy thay thế bằng lệnh puts.
5. Giá trị 1234 (số) có ý nghĩa khác hoàn toàn với giá trị "1234" (chuỗi).

### 2.1. Bài tập 1

1. **Thời lượng:** 10 phút.
2. **Mô tả bài toán:** Nhập vào một chuỗi liên tục không có khoảng trắng. Viết chương trình thực hiện các yêu cầu sau:
  - **Yêu cầu 1:** In từng ký tự trên từng hàng.
  - **Yêu cầu 2:** In từng ký tự theo trình tự ngược trên từng hàng.
3. **Gợi ý:**
  - Sử dụng hàm strlen để lấy độ dài của chuỗi.
  - **Yêu cầu 1:** Duyệt từng phần tử trong chuỗi từ đầu đến độ dài chuỗi-1 và in ra ký tự trong mỗi lần duyệt.
  - **Yêu cầu 2:** Duyệt ngược từng phần tử từ độ dài -1 về đầu chuỗi và in ra ký tự trong mỗi lần duyệt.

### 2.2. Bài tập 2

1. **Thời lượng:** 10 phút.
2. **Mô tả bài toán:** Nhập vào một chuỗi ký tự. Viết chương trình thực hiện các yêu cầu sau:
  - **Yêu cầu 1:** Chuyển tất cả ký tự từ IN HOA sang thường.
  - **Yêu cầu 2:** Chuyển tất cả ký tự từ thường sang IN HOA.

3. Gợi ý:

- Sử dụng hàm `strlen` để lấy độ dài của chuỗi.
- Duyệt từng phần tử trong chuỗi từ đầu đến độ dài chuỗi - 1.
  - **Yêu cầu 1:** Kết hợp hàm `toupper` khi duyệt mảng.
  - **Yêu cầu 2:** Kết hợp hàm `tolower` khi duyệt mảng.

2.3. Bài tập 3

1. Thời lượng: 15 phút.

2. Mô tả bài toán: Nhập vào một chuỗi ký tự. Viết chương trình thực hiện các yêu cầu sau:

- **Yêu cầu 1:** Đếm tất cả ký tự trong chuỗi (không tính khoảng trắng).
- **Yêu cầu 2:** Đếm tất cả ký số trong chuỗi.
- **Yêu cầu 3:** Đếm tất cả khoảng trắng trong chuỗi.

3. Gợi ý:

- **Yêu cầu 1:** Kết hợp hàm `isalpha` để kiểm tra ký tự khi duyệt mảng.
- **Yêu cầu 2:** Kết hợp hàm `isdigit` để kiểm tra ký tự khi duyệt mảng.
- **Yêu cầu 3:** Kết hợp hàm `isspace` để kiểm tra ký tự khi duyệt mảng.

2.4. Bài tập 4

1. Thời lượng: 15 phút.

2. Mô tả bài toán: Nhập vào hai chuỗi ký tự `s1` và `s2`. Viết chương trình thực hiện các yêu cầu sau:

- **Yêu cầu 1:** So sánh độ dài hai chuỗi và kết luận.
- **Yêu cầu 2:** Kiểm tra chuỗi `s2` có tồn tại trong chuỗi `s1` hay không và kết luận.
- **Yêu cầu 3:** Nối chuỗi `s2` vào chuỗi `s1` và in ra kết quả.

3. Gợi ý:

- **Yêu cầu 1:** Sử dụng hàm `strcmp`.
- **Yêu cầu 2:** Sử dụng hàm `strstr`.
- **Yêu cầu 3:** Sử dụng hàm `strcat`.

2.5. Bài tập 5

1. Thời lượng: 15 phút.

2. Mô tả bài toán: Nhập vào hai chuỗi ký tự `s1` và `s2`. Liệt kê các ký tự xuất hiện trong cả 2 chuỗi `s1` và `s2`.

3. Gợi ý:

- Xây dựng hàm **KiemTraTonTai** truyền vào 2 tham số ký tự cần kiểm tra và chuỗi cần kiểm tra. Hàm trả về 1 khi tìm thấy ký tự trong chuỗi. Ngược lại, trả về 0.
- So sánh độ dài 2 chuỗi. Lấy chuỗi ngắn hơn để duyệt mảng. Chuỗi dài là tham số trong hàm **KiemTraTonTai**.
- Trong quá trình duyệt mảng, gọi hàm **KiemTraTonTai** truyền tham số còn lại là phần tử ký tự trong chuỗi ngắn. Nếu hàm trả về 1 thì in ra màn hình.

## 2.6. Bài tập 6

1. **Thời lượng:** 15 phút.
2. **Mô tả bài toán:** Nhập vào một chuỗi ký tự s. Viết chương trình đếm số lượng từ có trong chuỗi s.
3. **Ví dụ:**
  - **Input chuỗi s:** “Tran Cong Nhat Phuong”
  - **Output:** Có 4 từ trong chuỗi “Tran Cong Nhat Phuong”.

## 2.7. Bài tập 7

1. **Thời lượng:** 15 phút.
2. **Mô tả bài toán:** Nhập vào một chuỗi ký tự s, và 1 ký tự c. Viết chương trình đếm số lượng ký tự c xuất hiện trong chuỗi s.
3. **Ví dụ:**
  - **Input chuỗi s:** “Tran Cong Nhat Phuong”.
  - **Input ký tự c:** ‘n’.
  - **Output:** Có 4 ký tự ‘N’ trong chuỗi “Tran Cong Nhat Phuong”.

## 2.8. Bài tập 8

1. **Thời lượng:** 15 phút.
2. **Mô tả bài toán:** Nhập vào họ tên của một người. Viết chương trình in ra tên viết tắt của người đó.
3. **Gợi ý:** In ra ký tự đầu tiên trong chuỗi. Duyệt mảng từ phần tử kế tiếp trong chuỗi, khi bắt gặp phần tử chứa ký tự khoảng trắng thì xuất ra phần tử kế sau. Sử dụng hàm toupper để in hoa ký tự.

## 2.9. Bài tập 9

1. **Thời lượng:** 30 phút.  
**Mô tả bài toán:** Nhập vào họ tên của một người. Viết chương trình tạo ra email của người đó dựa vào quy tắc sau:
  - **Quy tắc 1:** tên.họ\_và\_tên\_lót@greenacademy.edu.vn.
  - **Quy tắc 2:** tên.ký\_tự\_viết\_tắt\_của\_họ\_và\_tên\_lót@greenacademy.edu.vn.
2. **Ví dụ:**
  - **Quy tắc 1: Input:** Tran Cong Nhat Phuong.  
**Output:** phuong.trancongnhat@greenacademy.edu.vn.
  - **Quy tắc 2: Input:** Tran Cong Nhat Phuong.  
**Output:** phuong.tcn@greenacademy.edu.vn.

## 2.10. Bài tập 10

1. **Thời lượng:** 15 phút.
2. **Mô tả bài toán:** Nhập vào 1 chuỗi số. Viết chương trình phân tách chuỗi số có dấu phân tách phần ngàn, phần triệu.
3. **Ví dụ:** **Input:** 12345678      ➔      **Output:** 12,345,678

### 2.11. Bài tập 11

1. **Thời lượng:** 15 phút.
2. **Mô tả bài toán:** Nhập vào một chuỗi ký tự. Viết chương trình cắt chuỗi với vị trí và số lượng ký tự muốn cắt do người dùng nhập vào.
  - Xây dựng hàm **CatChuoi** truyền vào 2 tham số vị trí và độ dài cắt. Hàm trả về chuỗi ký tự đã cắt.
  - Tại chương trình chính, yêu cầu người dùng nhập chuỗi ký tự, vị trí và độ dài chuỗi muốn cắt. Gọi hàm **CatChuoi** và hiển thị kết quả sau khi cắt.
3. **Ví dụ:**
  - **Input:** s = “Tran Cong Nhat Phuong”, vị trí bắt đầu cắt = 10, độ dài cắt = 11.
  - **Output:** Nhat Phuong.

### 2.12. Bài tập 12

1. **Thời lượng:** 45 phút.
2. **Mô tả bài toán:** Viết chương trình đăng ký tài khoản học viên. Thông tin đăng ký bao gồm: Email, mật khẩu, xác nhận mật khẩu.
  - a) Xây dựng chương trình con:
    - Hàm **EmailValidation** truyền vào tham số email. Hàm kiểm tra tính hợp lệ của email bằng việc xem xét chuỗi email có chứa ký tự '@' và '.' hay không? Nếu có, hàm trả về giá trị 1. Ngược lại, hàm trả về giá trị 0.
    - Hàm **PasswordValidation** truyền vào tham số mật khẩu. Hàm kiểm tra tính hợp lệ của password bằng việc xem xét chuỗi password có chứa ít nhất 8 ký tự, trong đó có bao gồm ký tự chữ thường, ký tự chữ hoa và ký tự đặc biệt hay không? Nếu có, hàm trả về giá trị 1. Ngược lại, hàm trả về giá trị 0.
    - Hàm **ConfirmValidation** truyền vào 2 tham số chuỗi mật khẩu và chuỗi xác nhận mật khẩu. Hàm trả về giá trị 1 khi hai chuỗi này giống nhau hoàn toàn. Ngược lại, hàm trả về giá trị 0.
  - b) Tại chương trình chính:
    - Khai báo 3 chuỗi email, password và xác nhận password.
    - Yêu cầu người dùng nhập Email và gọi hàm **EmailValidation** để kiểm tra tính hợp lệ của Email. Hàm trả về giá trị 0 thì yêu cầu người dùng nhập lại.
    - Yêu cầu người dùng nhập Password và gọi hàm **PasswordValidation** để kiểm tra tính hợp lệ của Password. Hàm trả về giá trị 0 thì yêu cầu người dùng nhập lại.
    - Yêu cầu người dùng nhập ConfirmPassword và gọi hàm **ConfirmValidation** để kiểm tra tính hợp lệ của ConfirmPassword. Hàm trả về giá trị 0 thì yêu cầu người dùng nhập lại từ bước nhập Password.
    - Xuất thông báo “*Register successfully*” khi tất cả thông tin đều hợp lệ.



## Câu hỏi ôn tập

(Xem đáp án ở trang )

3.1. Trong các phương án dưới đây, đâu là một chuỗi ký tự?

- A. Static String
- B. "Static String"
- C. 'Static String'
- D. char string[100];

3.2. Ký tự nào được sử dụng để kết thúc chuỗi?

- A. '.'
- B. ''
- C. '\0'
- D. '/0'

3.3. Hàm nào được sử dụng để so sánh 2 chuỗi ký tự?

- A. ==
- B. compare();
- C. stringcompare();
- D. strcmp();

3.4. Hàm nào được sử dụng để thêm một chuỗi vào tận cùng của một chuỗi khác?

- A. append();
- B. stringadd();
- C. strcat();
- D. stradd();

3.5. Hãy cho biết kết quả của đoạn code sau:

```
#include <stdio.h>
#include <conio.h>
void main()
{
    char c=65;
    c = c + 10;
    printf("%d", c);
    getch();
}
```

- A. A
- B. K
- C. 65
- D. 75

3.6. Hãy cho biết kết quả của đoạn code sau:

```
#include <stdio.h>
#include <conio.h>
int main()
{
    char str[] = "GREENACADEMY\0\0.EDU\0";
    printf("%s\n", str);
    getch();
}
```

- A. GREENACADEMY
- B. GREENACADEMY\0
- C. GREENACADEMY\0.EDU
- D. GREENACADEMY\0.EDU \0

3.7. Cách khai báo nào sau đây là không đúng?

- A. int i = 0016;
- B. char c = "a";
- C. int d = 0x4b;
- D. char c[] = "a";

3.8. Dữ liệu ký tự (character) bao gồm:

- A. Các ký tự chữ số (digit).
- B. Các ký tự chữ cái (alpha).
- C. Các ký tự dấu câu (punctuation).
- D. Tất cả đều đúng

3.9. Trong các hàm sau, hàm nào để nhập một ký tự từ bàn phím ngay sau khi gõ, không chờ nhấn Enter và tự động hiện ra màn hình?

- A. scanf();
- B. getchar();
- C. getch();
- D. getche();

3.10. Trong các hàm sau, hàm nào để nhập một ký tự từ bàn phím ngay sau khi gõ, không chờ nhấn Enter và không hiện ra màn hình?

- A. scanf();
- B. getchar();
- C. getch();
- D. getche();

3.11. Cho các khai báo `char tb, mang[15];`. Hãy cho biết câu lệnh nào dưới đây là đúng?

- A. `tb = "Hello";`
- B. `gets(mang);`
- C. `mang = "Hello";`
- D. `gets(tb);`

3.12. Khi nhập vào nội dung *"Good morning, everyone"*. Hãy cho biết kết quả của đoạn code sau:

```
#include <stdio.h>
#include <conio.h>
void main()
{
    char str[80];
    printf("Input something: ");
    scanf("%s", str);
    printf("%s", str);
    getch();
}
```

- A. *Good morning, everyone.*
- B. *Good morning.*
- C. *Good.*
- D. Lỗi Runtime Error.