# Các Hàm memset() - memcpy() - memcmp() Trong C

Chức năng : Giới thiệu về các hàm thao tác trên bộ nhớ trong thư viện string.h trong C++.

## memset()

Cú pháp của hàm như sau:

void \* memset(void \* ptr, int value, size\_t num);

Tác dụng của hàm là set num byte giá trị của vùng nhớ tính từ vị trí ptr trỏ tới bằng giá trị value

Ví dụ:

#include <stdio.h>

#include <string.h>

int main()

{

char str[30] = "We are Stdio's Authors";

printf("%s\n", str);

memset(str + 3, '@', 10); // set 10 byte từ vị trí thứ 4 của str thành @

printf("%s\n", str);

return 0;

}

Kết quả in ra màn hình sẽ là:

We are Stdio's Authors

We @@@@@@@@@@s Authors

Tuy memset() có công dụng khá hay, nhưng ít khi lập trình viên sử dụng đến, trừ trường hợp muốn set toàn bộ vùng nhớ thành 1 giá trị gì đó, thường là giá trị 0.

## memcpy()

Cú pháp của hàm như sau:

void \* memcpy(void \* destination, const void \* source, size\_t num);

Tác dụng của hàm là sao chép num byte từ vị trí mà source trỏ tới đến vị trí mà destination trỏ tới

Có ví dụ sau:

#include <stdio.h>

#include <string.h>

int main()

{

char str1[30] = "We are Stdio's Authors";

char str2[30] = "World";

memcpy(str1 + 2, str2, 5);

printf("%s\n", str1);

return 0;

}

Kết quả in ra màn hình như sau:

WeWorldStdio's Authors

Tác dụng của hàm này khác với strcpy(), strcpy() sao chép vùng nhớ từ vị trí mà source trỏ tới cho tới vị trí của kí tự ‘\0’ tới destination, do đó destination có giá trị chuỗi giống với source, còn với memcpy() thì việc sao chép như thế nào hoàn toàn do người dùng kiểm soát. Tùy trường hợp mà sử dụng 2 hàm này.

## memcmp()

Cú pháp của hàm này như sau:

int memcmp(const void \* ptr1, const void \* ptr2, size\_t num);

Tác dụng: so sánh giá trị các vùng nhớ mà ptr1 và ptr2 trỏ tới theo từng byte, sẽ dừng lại khi so sánh đủ num byte. Khác với strcmp() luôn dừng lại khi gặp kí tự ‘\0’.

Giá trị trả về:

* -1: Khi giá trị của byte đầu tiên không trùng của ptr1 nhỏ hơn ptr2.
* 0: Khi giá trị của 2 vùng nhớ bằng nhau
* 1: Khi giá trị của byte đầu tiên không trùng của ptr1 lớn hơn ptr2.

Về độ lớn giá trị của các kí tự có thể tham khảo bài viết [Bảng Mã ASCII](https://www.stdio.vn/computer-arc/bang-ma-ascii-slmLl1).