

[Dashboard](#) / [My courses](#) / [SFP:HK2-2019-2020](#) / [Chuỗi ký tự](#) / [Bài tập chuỗi ký tự](#)

Started on	Tuesday, 2 June 2020, 9:25 PM
State	Finished
Completed on	Friday, 5 June 2020, 8:19 PM
Time taken	2 days 22 hours
Marks	5.67/6.00
Grade	9.45 out of 10.00 (94%)

Question **1**

Correct

Mark 1.00 out of 1.00

Viết 1 [hàm](#) cho phép chèn ký tự c vào vị trí p trong chuỗi st

Khuôn dạng (Prototype)

- Tên [hàm](#) (function name): **Insert()**
- Tham số (parameters):
 - c** - kiểu char
 - p** - kiểu int
 - st[]** - kiểu char
- Kiểu trả về (return type): **void**

Thân hàm (Body)

- Cho phép chèn ký tự c vào vị trí p trong chuỗi st

Chú ý

- Giá trị của các tham số luôn hợp lệ, không cần kiểm tra.
- Chỉ viết [hàm](#), KHÔNG VIẾT TOÀN BỘ CHƯƠNG TRÌNH
- Xem thêm chi tiết trong phần **For example**.

Gợi ý

- Ý tưởng là dời các ký tự từ vị trí p trở đi ra sau 1 vị trí, sau đó đưa c vào vị trí p
- Giải thuật như sau:
 - n = Chiều dài chuỗi st
 - $\forall i = \overline{n - 1, p}$
 - $st[i + 1] = st[i]$
 - $st[p] = c;$
 - Số ký tự tăng lên $n = n + 1$
 - Đánh dấu hết chuỗi bằng cách đặt $st[n] = '\0'$

For example:

Test	Input	Result
char str[500]; fgets(str,500,stdin); printf("%s",str); Insert(' ', 3,str); printf("%s",str);	cantho	cantho can tho
char str[500]; fgets(str,500,stdin); printf("%s",str); Insert('-', 4,str); printf("%s",str);	CNTTDHCT	CNTTDHCT CNTT-DHCT

Answer: (penalty regime: 33.3, 66.7, ... %)

```
1 #include <stdio.h>
2 #include <string.h>
3 void Insert(char c, int p, char st[]){
4     int n;
5     n = strlen(st);
6     int i;
7     for(i = n-1; i > p-1; i--){
8         st[i+1] = st[i];
9     }
10    st[p] = c;
11    n++;
12    st[n] = '\0';
13 }
```

	Test	Input	Expected	Got	
✓	<pre>char str[500]; fgets(str,500,stdin); printf("%s",str); Insert(' ', 3,str); printf("%s",str);</pre>	cantho	cantho can tho	cantho can tho	✓
✓	<pre>char str[500]; fgets(str,500,stdin); printf("%s",str); Insert('-', 4,str); printf("%s",str);</pre>	CNTTDHCT	CNTTDHCT CNTT-DHCT	CNTTDHCT CNTT-DHCT	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question **2**

Correct

Mark 1.00 out of 1.00

Viết 1 [hàm](#) cho phép xóa ký tự ở vị trí p trng chuỗi st

Khuôn dạng (Prototype)

- Tên [hàm](#) (function name): **Delete()**
- Tham số (parameters):
 - p** - kiểu int
 - st[]** - kiểu char
- Kiểu trả về (return type): **void**

Thân hàm (Body)

- Cho phép xóa ký tự ở vị trí p trong chuỗi st

Chú ý

- Giá trị của các tham số luôn hợp lệ, không cần kiểm tra.
- Chỉ viết [hàm](#), KHÔNG VIẾT TOÀN BỘ CHƯƠNG TRÌNH
- Xem thêm chi tiết trong phần **For example**.

Gợi ý

- Ý tưởng là dời các ký tự từ vị trí p trở đi ra trước 1 vị trí
- Giải thuật như sau:

- n = Chiều dài chuỗi st

- $\forall i = \overline{p, n - 2}$

$st[i] = st[i + 1]$

- Số ký tự giảm đi n=n-1

- Đánh dấu hết chuỗi bằng cách đặt $st[n] = '\0'$

For example:

Test	Input	Result
char str[500]; fgets(str,500,stdin); printf("%s",str); Delete(3,str); printf("%s",str);	can tho	can tho cantho
char str[500]; fgets(str,500,stdin); printf("%s",str); Delete(4,str); printf("%s",str);	CNTT DHCT	CNTT DHCT CNTTDHCT

Answer: (penalty regime: 33.3, 66.7, ... %)

```
1 #include <stdio.h>
2 #include <string.h>
3 void Delete(int p, char st[]){
4     int n,i;
5     n = strlen(st);
6     for(i = p; i < n-1; i++){
7         st[i] = st[i+1];
8     }
9     n--;
10    st[n] = '\0';
11 }
```

	Test	Input	Expected	Got	
✓	<pre>char str[500]; fgets(str,500,stdin); printf("%s",str); Delete(3,str); printf("%s",str);</pre>	can tho	can tho cantho	can tho cantho	✓
✓	<pre>char str[500]; fgets(str,500,stdin); printf("%s",str); Delete(4,str); printf("%s",str);</pre>	CNTT DHCT	CNTT DHCT CNTTDHCT	CNTT DHCT CNTTDHCT	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question **3**

Correct

Mark 1.00 out of 1.00

Viết 1 [hàm](#) cho phép lấy chuỗi con từ vị trí pos, lấy number ký tự của một chuỗi đầu vào cho trước

Khuôn dạng (Prototype)

- Tên [hàm](#) (function name): **subString()**
- Tham số (parameters):
 - s[]** - kiểu char
 - pos** - kiểu int
 - number** - kiểu in
- Kiểu trả về (return type): **char***

Thân [hàm](#) (Body)

- Cho phép lấy về chuỗi con từ vị trí pos, lấy number ký tự của chuỗi đầu vào; kết quả trả về của [hàm](#) là con trỏ chỉ tới chuỗi kết quả

Chú ý

- Giá trị của các tham số luôn hợp lệ, không cần kiểm tra.
- Chỉ viết [hàm](#), KHÔNG VIẾT TOÀN BỘ CHƯƠNG TRÌNH
- Xem thêm chi tiết trong phần **For example**.

Chú ý

- Khai báo một con trỏ ký tự des
- Cấp phát 1 vùng nhớ mới có (number + 1) ký tự, cho des chỉ tới địa chỉ của vùng nhớ được cấp phát này
- Lặp từ vị trí pos tới pos + number trong chuỗi nguồn để đưa từng ký tự sang chuỗi des.
- Thêm ký tự kết thúc chuỗi vào des
- Trả về kết quả cho [hàm](#)

For example:

Test	Result
char st[]="toi di hoc"; char *res = subString(st, 4,2); puts(st); puts(res); free(res);	toi di hoc di
char st[]="Dai Hoc Can Tho"; char *res = subString(st, 0,7); puts(st); puts(res); free(res);	Dai Hoc Can Tho Dai Hoc

Answer: (penalty regime: 33.3, 66.7, ... %)

```
1  #include <stdio.h>
2  #include <string.h>
3  #include <stdlib.h>
4  char *subString(char s[], int pos, int number){
5      char *des;
6      des = (char *) malloc(number + 1);
7      int i,j = 0;
8      for(i = pos; i < number + pos; i++){
9          *(des+j) = s[i];
10         j++;
11     }
12     *(des+j) = '\0';
13     return des;
14
15 }
```



	Test	Expected	Got	
✓	<pre>char st[]="toi di hoc"; char *res = subString(st, 4,2); puts(st); puts(res); free(res);</pre>	toi di hoc di	toi di hoc di	✓
✓	<pre>char st[]="Dai Hoc Can Tho"; char *res = subString(st, 0,7); puts(st); puts(res); free(res);</pre>	Dai Hoc Can Tho Dai Hoc	Dai Hoc Can Tho Dai Hoc	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question **4**

Correct

Mark 1.00 out of 1.00

Viết chương trình đếm số từ trong một chuỗi đầu vào. Biết khoảng trắng là ký hiệu phân cách mỗi từ.

Đầu vào

- Chuỗi đầu vào có tối đa 500 ký tự

Đầu ra

- Dòng đầu tiên là chuỗi nhập
- Dòng thứ 2 là 1 số nguyên ghi số từ trong chuỗi nhập

Ghi chú:

- Dữ liệu đầu vào là hợp lệ.

Xem phần ví dụ để biết chi tiết về đầu vào/đầu ra.

For example:

Input	Result
tOI dI HOC	tOI dI HOC 3
Dai Hoc Can Tho	Dai Hoc Can Tho 4
AVA	AVA 1

Answer: (penalty regime: 33.3, 66.7, ... %)

```
1  #include <stdio.h>
2  #include <string.h>
3  int main(){
4      char s[500];
5      fgets(s,500,stdin);
6      int n;
7      n = strlen(s);
8      if(s[n-1]=='\n'){
9          n--;
10         s[n] = '\0';
11     }
12     puts(s);
13     int i,count = 1;
14     for(i = 0; i < n; i++){
15         if(s[i]==' '){
16             count++;
17             if(s[i] == ' ' && (s[i-1] == ' ')){
18                 count--;
19             }
20         }
21     }
22     printf("%d",count);
23     return 0;
24
25 }
```

	Input	Expected	Got	
✓	tOI dI HOC	tOI dI HOC 3	tOI dI HOC 3	✓
✓	Dai Hoc Can Tho	Dai Hoc Can Tho 4	Dai Hoc Can Tho 4	✓
✓	AVA	AVA 1	AVA 1	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question **5**

Correct

Mark 1.00 out of 1.00

Viết [hàm](#) copy nội dung của chuỗi này vào chuỗi kia.

Khuôn dạng (Prototype)

- Tên [hàm](#) (function name): **mystrcpy**
- Tham số (parameters):
 - char *s1**: chuỗi gốc
 - char *s2**: chuỗi copy
- Kiểu trả về (return type): **void**

Thân hàm (Body)

- Copy nội dung chuỗi s1 vào chuỗi s2. giả sử chuỗi s2 có đủ chỗ để chứa toàn bộ nội dung chuỗi s1.

Chú ý

- Giá trị của các tham số luôn hợp lệ, không cần kiểm tra.
- Chỉ viết [hàm](#), KHÔNG VIẾT TOÀN BỘ CHƯƠNG TRÌNH
- Xem thêm chi tiết trong phần **For example**.

Gợi ý

- Dùng vòng lặp copy từng ký tự, kể cả ký tự kết thúc chuỗi '\0'

For example:

Test	Result
char c[100]; mystrcpy("Hello World", c); printf("%s", c);	Hello World
char *p = "Hello World"; char *c = (char*)malloc(100); mystrcpy(p, c); printf("%s", c); free(c);	Hello World

Answer: (penalty regime: 33.3, 66.7, ... %)

```
1 #include <stdio.h>
2 #include <string.h>
3 #include <stdlib.h>
4 void mystrcpy(char *s1, char *s2){
5     int i = 0;
6     while(*(s1+i) != '\0'){
7         *(s2+i) = *(s1+i);
8         i++;
9     }
10    *(s2+i) = '\0';
11 }
```

	Test	Expected	Got	
✓	char c[100]; mystrcpy("Hello World", c); printf("%s", c);	Hello World	Hello World	✓
✓	char c[100] = "AAAAAAAAAAAAAAAAAAAAAAAAA"; mystrcpy("Hello World", c); printf("%s", c);	Hello World	Hello World	✓

	Test	Expected	Got	
✓	<pre>char *p = "Hello World"; char *c = (char*)malloc(100); mystrcpy(p, c); printf("%s", c); free(c);</pre>	Hello World	Hello World	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question **6**

Correct

Mark 0.67 out of 1.00

Viết 1 [hàm](#) so sánh 2 chuỗi đầu vào (không phân biệt hoa thường). Lưu ý không sử dụng [hàm](#) strcmp() của thư viện string.h

Khuôn dạng (Prototype)

- Tên [hàm](#) (function name): **Compare()**
- Tham số (parameters):
 - s1[]** - kiểu char
 - s2[]** - kiểu char
- Kiểu trả về (return type): **int**

Thân [hàm](#) (Body)

- Nếu s1 < s2: kết quả trả về là một số nguyên âm nào đó
- Nếu s1 > s2: kết quả trả về là một số nguyên dương nào đó
- Nếu s1 = s2: kết quả trả về là 0

Chú ý

- Giá trị của các tham số luôn hợp lệ, không cần kiểm tra.
- Chỉ viết [hàm](#), KHÔNG VIẾT TOÀN BỘ CHƯƠNG TRÌNH
- Xem thêm chi tiết trong phần **For example**.

Gợi ý:

- Áp dụng giải thuật duyệt qua các chuỗi đầu vào s mà không sử dụng [hàm](#) strcmp() của string.h. Ở đó lấy từng cặp ký tự trong cả 2 chuỗi ở cùng 1 vị trí trừ cho nhau, nếu gặp vị trí sai khác đầu tiên thì dừng lại.
- Chú ý vấn đề không phân biệt hoa thường
- Giải thuật cơ bản như sau:

i = 0

j=0

d=0

Lặp trong khi (ký tự $s1_i \neq '0'$ AND ký tự $s2_j \neq '0'$ AND !d)

+ Tính hiệu d = $s1_i - s2_j$

+ i=i+1

+ j=j+1

Biện luận để trả về kết quả cho [hàm](#)

For example:

Test	Input	Result
<pre>char s1[500], s2[500]; int len; fgets(s1,500,stdin); //Remove '\n' in fgets len = strlen(s1); if (s1[len-1]=='\n') { len--; s1[len]='\0'; } fgets(s2,500,stdin); //Remove '\n' in fgets len = strlen(s2); if (s2[len-1]=='\n') { len--; s2[len]='\0'; } if (Compare(s1,s2) < 0) printf("1 < 2"); else if (Compare(s1,s2) > 0) printf("1 > 2"); else printf("1 = 2");</pre>	<pre>lam xung Lam Xung</pre>	<pre>1 = 2</pre>

Test	Input	Result
<pre>char s1[500], s2[500]; int len; fgets(s1,500,stdin); //Remove '\n' in fgets len = strlen(s1); if (s1[len-1]=='\n') { len--; s1[len]='\0'; } fgets(s2,500,stdin); //Remove '\n' in fgets len = strlen(s2); if (s2[len-1]=='\n') { len--; s2[len]='\0'; } if (Compare(s1,s2) < 0) printf("1 < 2"); else if (Compare(s1,s2) > 0) printf("1 > 2"); else printf("1 = 2");</pre>	Cntt K44 CNTT K45	1 < 2

Answer: (penalty regime: 33.3, 66.7, ... %)

1	#include <stdio.h>	
2	#include <string.h>	
3	int Compare(char s1[], char s2[]){	
4	int j=0,i=0,d=0;	
5	while(s1[i] != '\0' && s2[j] != '\0'){	
6	d = s1[i] - s2[j];	
7	i++;	
8	j++;	
9	}	
10	return d;	
11	}	

	Test	Input	Expected	Got	
✓	<pre>char s1[500], s2[500]; int len; fgets(s1,500,stdin); //Remove '\n' in fgets len = strlen(s1); if (s1[len-1]=='\n') { len--; s1[len]='\0'; } fgets(s2,500,stdin); //Remove '\n' in fgets len = strlen(s2); if (s2[len-1]=='\n') { len--; s2[len]='\0'; } if (Compare(s1,s2) < 0) printf("1 < 2"); else if (Compare(s1,s2) > 0) printf("1 > 2"); else printf("1 = 2");</pre>	lam xung Lam Xung	1 = 2	1 = 2	✓

	Test	Input	Expected	Got	
✓	<pre>char s1[500], s2[500]; int len; fgets(s1,500,stdin); //Remove '\n' in fgets len = strlen(s1); if (s1[len-1]=='\n') { len--; s1[len]='\0'; } fgets(s2,500,stdin); //Remove '\n' in fgets len = strlen(s2); if (s2[len-1]=='\n') { len--; s2[len]='\0'; } if (Compare(s1,s2) < 0) printf("1 < 2"); else if (Compare(s1,s2) > 0) printf("1 > 2"); else printf("1 = 2");</pre>	Cntt K44 CNTT K45	1 < 2	1 < 2	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00. Accounting for previous tries, this gives **0.67/1.00**.

[◀ Một số vấn đề cơ bản trong lập trình](#)

Jump to...

[Tóm tắt về chuỗi ký tự ▶](#)