

Status	Finished
Started	Sunday, 12 January 2025, 3:15 AM
Completed	Sunday, 12 January 2025, 3:15 AM
Duration	4 secs
Marks	0.00/140.00
Grade	0.00 out of 10.00 (0%)

Question 1

Not answered

Mark 0.00 out of 10.00

[RightAdding]

Viết hàm `void pad_right(char *s, int n)` với nhiệm vụ điền thêm các kí tự '_' vào cuối chuỗi `s` để `s` có độ dài bằng `n`. Nếu chuỗi `s` có độ dài lớn `n` thì không phải làm gì cả.

Ví dụ, hàm `pad_right` nhận vào chuỗi `s = "abc"` và `n = 5` thì khi kết thúc hàm `pad_right`, chuỗi `s` có giá trị là `"abc_"`

Answer: (penalty regime: 0 %)

1

Question 2

Not answered

Mark 0.00 out of 10.00

[LeftAdding]

Viết hàm `void pad_left(char *s, int n)` với nhiệm vụ điền thêm các ký tự '_' vào đầu chuỗi s để s có độ dài bằng n . Nếu chuỗi s có độ dài lớn n thì không phải làm gì cả.

Ví dụ, hàm `pad_left` nhận vào chuỗi $s = \text{"abc"}$ và $n = 5$ thì khi kết thúc hàm `pad_left`, chuỗi s có giá trị là `"__abc"`

Answer: (penalty regime: 0 %)

1	
---	--

Question 3

Not answered

Mark 0.00 out of 10.00

[trim_right]

Viết hàm `void trim_right(char *s)` với nhiệm vụ xóa các kí tự trắng đứng ở cuối chuỗi `\(s\)` nếu có.

Ví dụ, hàm `trim_right` nhận vào chuỗi `\(s=\)` "abc " thì sau khi kết thúc hàm `trim_right`, chuỗi `\(s\)` sẽ có giá trị là "abc"

Answer: (penalty regime: 0 %)

1

Question 4

Not answered

Mark 0.00 out of 10.00

[Trim_Left]

Viết hàm `void trim_left(char *s)` với nhiệm vụ xóa các kí tự trắng đứng ở đầu chuỗi `\s\` nếu có.

Ví dụ, hàm `trim_left` nhận vào chuỗi `\s=\` " abc" thì sau khi kết thúc hàm `trim_left`, chuỗi `\s\` sẽ có giá trị là "abc"

Answer: (penalty regime: 0 %)

1	
---	--

Question 5

Not answered

Mark 0.00 out of 10.00

[Reverse]

Viết một hàm `void reverse(char *s)` nhận vào một chuỗi và đảo ngược thứ tự của chuỗi s đó.

Ví dụ, hàm `reverse` nhận vào chuỗi `\s) = "abc"`. Khi kết thúc hàm `reverse` chuỗi `\s)` sẽ có giá trị là `"cba"`.

Answer: (penalty regime: 0 %)

1	
---	--

Question 6

Not answered

Mark 0.00 out of 10.00

[GetSum]

Viết hàm `int getSum(int *a, int n)` nhận vào mảng số nguyên `a` có `n` phần tử và tính tổng giá trị của các phần tử trong mảng `a`.

Answer: (penalty regime: 0 %)

1	
---	--

Question 7

Not answered

Mark 0.00 out of 10.00

[Delete_Char]

Viết hàm `void delete_char(char *S, char c)` nhận vào chuỗi `\(S\)` và ký tự `\(c\)`, sau đó hàm này sẽ xóa hết các ký tự có giá trị bằng `\(c\)` trong chuỗi `\(S\)`.

Ví dụ, hàm `delete_char` nhận vào chuỗi `\(S=\)` "abcdac" và ký tự `\(c=\)` 'a' thì sau khi kết thúc hàm, `\(S\)` sẽ nhận giá trị là "bcdac".

For example:

Input	Result
abcccaa a	bccc

Answer: (penalty regime: 0 %)

1	
---	--

Question 8

Not answered

Mark 0.00 out of 10.00

[Merge]

Viết hàm `int* merge(int* firstArr, int lenArr1, int* secondArr, int lenArr2)` thực hiện việc nối hai mảng số nguyên đã sắp xếp với độ dài biết trước thành một mảng số nguyên duy nhất và thứ tự sắp xếp không đổi.

Hàm nhận đầu vào là hai mảng số nguyên đã sắp xếp `\(firstArr\)` và `\(secondArr\)`, với độ dài mảng lần lượt là `\(lenArr1\)` và `\(lenArr2\)`.

Hàm trả về mảng số nguyên là kết quả của việc nối hai mảng đầu vào thành mảng duy nhất và các phần tử trong mảng cũng được sắp xếp theo thứ tự không đổi (tăng dần hoặc giảm dần) như các mảng đầu vào.

Lưu ý: các mảng đầu vào có thể [sắp xếp tăng dần](#) hoặc giảm dần.

For example:

Input	Result
8 0 5 5 15 23 25 33 61	0 5 5 12 15 23 25 25 33 36 44 45 45 53 61 65 71
9 12 25 36 44 45 45 53 65 71	

Answer: (penalty regime: 0 %)

1

Question 9

Not answered

Mark 0.00 out of 10.00

[GetDataFromPointer]

Viết chương trình nhập vào từ bàn phím một số nguyên bằng con trỏ và in lại giá trị số nguyên đó ra màn hình.

For example:

Input	Result
6	6

Answer: (penalty regime: 0 %)

Reset answer

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main()
6 {
7     int *pointer;
8     // your code here
9     cout << *pointer;
10    return 0;
11 }
```

Question 10

Not answered

Mark 0.00 out of 10.00

[getPointerToTen]

Viết hàm `int* getPointerToTen()` . Hàm này khai báo một con trỏ kiểu nguyên, cấp phát bộ nhớ của một số nguyên cho con trỏ đó và gán giá trị 10 cho vùng bộ nhớ đó. Hàm trả về con trỏ đã khai báo.

Answer: (penalty regime: 0 %)

1	
---	--

Question 11

Not answered

Mark 0.00 out of 10.00

[PrintImage]

Một ảnh xám có thể biểu diễn bởi một ma trận hai chiều, trong đó mỗi phần tử của mảng biểu diễn một điểm ảnh (có giá trị từ 0 - 255). Giả sử ta muốn in một ảnh xám ra màn hình dòng lệnh, ta có thể in giá trị của từng điểm ảnh. Viết hàm `void printImage(int** img, int height, int width)` nhận tham số truyền vào là một con trỏ trỏ đến [mảng hai chiều](#) biểu diễn ảnh, chiều cao và chiều rộng của ảnh. Hàm in ra màn hình giá trị của từng điểm ảnh theo từng hàng, trên một hàng các giá trị luôn chiếm \4\ khoảng trống.

Gợi ý: sử dụng `cout << setw(4) << number;` (C++) hoặc `printf("%4d", number);` (C/C++).

Answer: (penalty regime: 0 %)

1	
---	--

Question 12

Not answered

Mark 0.00 out of 10.00

[SelfDividingNumbers]

Số tự phân chia là số chia hết cho mỗi chữ số của nó.

Ví dụ: 128 là số tự phân chia vì $128 \% 1 == 0$, $128 \% 2 == 0$ và $128 \% 8 == 0$.

Ngoài ra, số tự phân chia không được phép chứa số 0.

Viết hàm `int* selfDividingNumbers(int left, int right, int* returnSize)` nhận đầu vào là một ngưỡng chặn dưới (left) và một ngưỡng chặn trên (right). Hàm thực hiện tính toán và trả về danh sách các số tự phân chia nằm trong đoạn giới hạn bởi left và right.

Giá trị biến (returnSize) đại diện cho số các số tự phân chia của mảng trả về và giá trị của biến có thể thay đổi trong hàm.

Ghi chú: Tránh phép chia 0.

For example:

Input	Result
37 58	44 48 55

Answer: (penalty regime: 0 %)

1

Question 13

Not answered

Mark 0.00 out of 10.00

[SquareWithPointer]

Viết hàm tính bình phương của một số thực.

Hàm `double* getSquare (double number)` nhận đầu vào là một số thực `(number)` và trả về con trỏ kiểu `(double)` chứa giá trị bình phương của số thực đã cho.

Answer: (penalty regime: 0 %)

1	
---	--

Question 14

Not answered

Mark 0.00 out of 10.00

[Transpose Matrix]

Ma trận chuyển vị là một [ma trận](#) ở đó các hàng được thay thế bằng các cột, và ngược lại.

Hãy viết hàm `int** transpose (int** matrix, int m, int n)` nhận đầu vào là một ma trận $(A \text{ } \backslash (matrix) \backslash)$ bất kỳ có kích cỡ $(m \text{ } \backslash times \text{ } n \backslash)$ chứa các giá trị nguyên.

Hàm trả về ma trận chuyển vị của ma trận $(A \backslash)$ là ma trận $(A^T \backslash)$ có kích thước $(n \text{ } \backslash times \text{ } m \backslash)$ với các hàng được thay thế bởi các cột.

For example:

Input	Result
4 3	11 5 9 5
11 0 3	0 5 5 5
5 5 4	3 4 8 1
9 5 8	
5 5 1	

Answer: (penalty regime: 0 %)

1	
---	--

[Back to Course](#)