

[Dashboard](#) / [My courses](#) / [SFP:HK2-2019-2020](#) / [Tập tin](#) / [Thao tác với tập tin văn bản](#)

Started on	Wednesday, 10 June 2020, 8:17 PM
State	Finished
Completed on	Saturday, 20 June 2020, 10:48 PM
Time taken	10 days 2 hours
Marks	5.33/11.00
Grade	4.85 out of 10.00 (48%)

Question **1**

Correct

Mark 0.00 out of 1.00

Kiểm tra 1 mảng có được sắp xếp tăng dần hay không?

Cho một mảng các số thực được lưu trữ trong tập tin văn bản như sau:

- Dòng đầu tiên ghi 1 số nguyên n là số phần tử của mảng
- Dòng thứ 2 ghi n số thực, mỗi số cách nhau khoảng trắng

Viết [hàm](#) checkSort() để kiểm tra mảng số thực trong tập tin đầu vào có tăng dần hay không? Nếu tăng dần hiển thị YES, ngược lại ghi NO

Khuôn dạng (Prototype)

- Tên [hàm](#): checkSort
- Tham số:
 - char *sInput: tên file đầu vào
- Kiểu trả về: void

Thân [hàm](#)

- Đọc nội dung file sInput, lưu trong mảng A.
- Kiểm tra mảng A tăng dần hay không? Nếu tăng dần hiển thị YES, ngược lại ghi NO

Gợi ý

- Dùng fscanf để đọc từ tập tin vào.
- Dữ liệu vào là hợp lệ, sinh viên không cần kiểm tra

Ví dụ file nguồn

5
1.0 2.0 3.0 5.0 10.0

In ra:

YES

For example:

Input	Result
tangdan1.txt	YES
tangdan2.txt	YES

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

```
1 void checkSort(char *sInput){
2     FILE *f;
3     int i,n,flag = 1;
4     float a[100];
5     f = fopen(sInput,"r");
6     fscanf(f,"%d", &n);
7     for(i = 0; i < n; i++){
8         fscanf(f,"%f", &a[i]);
9     }
10    fclose(f);
11    for(i = 0; i < n; ++i){
12        if(a[i-1] > a[i]){
13            flag = 0;
14            break;
15        }
16    }
17    if(flag)
18        printf("YES");
19    else
20        printf("NO");
21 }
```

	Input	Expected	Got	
--	-------	----------	-----	--

	Input	Expected	Got	
✓	tangdan1.txt	YES	YES	✓
✓	tangdan2.txt	YES	YES	✓
✓	tangdan3.txt	YES	YES	✓
✓	tangdan4.txt	NO	NO	✓
✓	tangdan5.txt	NO	NO	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00. Accounting for previous tries, this gives **0.00/1.00**.

Question **2**

Correct

Mark 0.00 out of 1.00

Mật mã Caesar là một phương pháp mã hoá cổ điển được đặt tên theo tên của Hoàng đế vĩ đại Julius Caesar. Tương truyền, Julius Caesar dùng phương pháp này để viết thư cho các người tình của mình.

Phương pháp mã hoá khá đơn giản: thay thế ký tự đang xét bằng ký tự đứng sau nó 3 ký tự trong bảng chữ cái, ví dụ: A được thay bằng D, B được thay bằng E, ..., X được thay bằng A, Y được thay bằng B, Z được thay bằng C.

Câu: THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG

Sẽ được mã hoá thành: WKH TXLFN EURZQ IRA MXPSV RYHU WKH ODCB GRJ

Viết **hàm encrypt** đọc 1 file văn bản, mã hoá nội dung của nó theo phương pháp mã hoá Caesar và ghi nội dung đã mã hoá vào một file văn bản khác.

Khuôn dạng (Prototype)

- Tên **hàm**: encrypt
- Tham số:
 - char *sInput: tên file cần mã hoá
 - char *sOutput: tên file kết quả
- Kiểu trả về: void

Thân hàm

- Đọc nội dung file sInput, mã hoá và ghi kết quả vào file sOutput.
- Quy tắc mã hoá như sau:
 - Chỉ mã hoá các chữ cái (A - Z, a - z), các ký tự khác giữ nguyên
 - Đổi các chữ cái thành chữ hoa và mã hoá chữ cái theo phương pháp mật mã Caesar.

Gợi ý

- Đọc từng dòng hoặc đọc từ ký tự, mã hoá, ghi kết quả ra file.
- Với từng ký tự là chữ cái A - Z chuyển nó về số 0 - 25, lấy số tương ứng cộng thêm 3 và chia cho 26 lấy phần dư (để đảm bảo các số nằm trong khoảng từ 0 - 25), chuyển số trở thành chữ cái lại A - Z.

For example:

Input	Result
file1.txt	WKH TXLFN EURZQ IRA MXPSV RYHU WKH ODCB GRJ

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

```
1 void encrypt(char *sInput, char *Output){
2     FILE *f,*fn;
3     char s[100];
4     int i;
5     f = fopen(sInput,"r");
6     fn = fopen(Output,"w");
7     while(fgets(s,100,f) != NULL){
8         if(s[strlen(s)-1] == '\n'){
9             s[strlen(s)-1] = '\0';
10        }
11        for(i = 0; i < strlen(s); i++){
12            if(s[i] >= 'A' && s[i] <= 'z'){
13                if(s[i] >= 'a'){
14                    s[i]-= 32;
15                }
16                if(s[i] >= 'X'){
17                    s[i]-=26;
18                }
19                s[i]+=3;
20            }
21        }
22        fputs(s,fn);
23        fprintf(fn,"\n");
24    }
25    fclose(f);
26    fclose(fn);
27 }
```

	Input	Expected	Got	
✓	file1.txt	WKH TXLFN EURZQ IRA MXPSV RYHU WKH ODCB GRJ	WKH TXLFN EURZQ IRA MXPSV RYHU WKH ODCB GRJ	✓

	Input	Expected	Got	
✔	file2.txt	<div>LI BRX IRUJHW PH:</div> <div>L ZDQW BRX WR NQRZ RQH WKLQJ.</div> <div>BRX NQRZ KRZ WKLV LV: LI L ORRN DW WKH FUBVWDO PRRQ, DW WKH UHG EUDQFK RI WKH VORZ DXWXPQ DW PB ZLQGRZ, LI L WRXFK QH DU WKH ILUH WKH LPSDOSDEOH DVK RU WKH ZULQNOHG ERGB RI WKH ORJ, HYHUBWKLQJ FDUULHV PH WR BRX, DV LI HYHUBWKLQJ WKDW HALVWV, DURPDV, OLJKW, PHWDOV, ZHUH OLWWOH ERDWV WKDW VDLO WRZDUG WKR VH LVOHV RI BRXUV WKDW ZDLW IRU PH.</div> <div>ZHOO, QRZ, LI OLWWOH EB OLWWOH BRX VWRS ORYLQJ PH L VKDOO VWRS ORYLQJ BRX OLWWOH EB OLWWOH.</div> <div>LI VXGGHQOB BRX IRUJHW PH GR QRW ORRN IRU PH, IRU L VKDOO DOUHDGB KDYH IRUJRWWHQ BRX.</div> <div>LI BRX WKLQN LW ORQJ DQG PDG, WKH ZLQG RI EDQQHUV WKDW SDVVHV WKURXJK PB OLIH, DQG BRX GHFLGH WR OHDYH PH DW WKH VKRUH RI WKH KHDUW ZKHUH L KDYH URRWV,</div> <div>UHPHPEHU WKDW RQ WKDW GDB, DW WKDW KRXU, L VKDOO OLIW PB DUPV DQG PB URRWV ZLOO VHW RII WR VHHN DQRWKHU ODQG.</div> <div>EXW LI HDFK GDB, HDFK KRXU, BRX IHHO WKDW BRX DUH GHVWLQHG IRU PH ZLWK LPSODFDEOH VZHHWQHVV, LI HDFK GDB D IORZHU FOLPEV XS WR BRXU OLSV WR VHHN PH, DK PB ORYH, DK PB RZQ, LQ PH DOO WKDW ILUH LV UHSHDWHG, LQ PH QRWKLQJ LV HAWLQJXLVKHG RU IRUJRWWHQ, PB ORYH IHGGV RQ BRXU ORYH, EHORYHG, DQG DV ORQJ DV BRX OLYH LW ZLOO EH LQ BRXU DUPV ZLWKRXW OHDYLQJ PLQH.</div> <div>EB SDEOR QHUXGD</div>	<div>LI BRX IRUJHW PH:</div> <div>L ZDQW BRX WR NQRZ RQH WKLQJ.</div> <div>BRX NQRZ KRZ WKLV LV: LI L ORRN DW WKH FUBVWDO PRRQ, DW WKH UHG EUDQFK RI WKH VORZ DXWXPQ DW PB ZLQGRZ, LI L WRXFK QH DU WKH ILUH WKH LPSDOSDEOH DVK RU WKH ZULQNOHG ERGB RI WKH ORJ, HYHUBWKLQJ FDUULHV PH WR BRX, DV LI HYHUBWKLQJ WKDW HALVWV, DURPDV, OLJKW, PHWDOV, ZHUH OLWWOH ERDWV WKDW VDLO WRZDUG WKR VH LVOHV RI BRXUV WKDW ZDLW IRU PH.</div> <div>ZHOO, QRZ, LI OLWWOH EB OLWWOH BRX VWRS ORYLQJ PH L VKDOO VWRS ORYLQJ BRX OLWWOH EB OLWWOH.</div> <div>LI VXGGHQOB BRX IRUJHW PH GR QRW ORRN IRU PH, IRU L VKDOO DOUHDGB KDYH IRUJRWWHQ BRX.</div> <div>LI BRX WKLQN LW ORQJ DQG PDG, WKH ZLQG RI EDQQHUV WKDW SDVVHV WKURXJK PB OLIH, DQG BRX GHFLGH WR OHDYH PH DW WKH VKRUH RI WKH KHDUW ZKHUH L KDYH URRWV,</div> <div>UHPHPEHU WKDW RQ WKDW GDB, DW WKDW KRXU, L VKDOO OLIW PB DUPV DQG PB URRWV ZLOO VHW RII WR VHHN DQRWKHU ODQG.</div> <div>EXW LI HDFK GDB, HDFK KRXU, BRX IHHO WKDW BRX DUH GHVWLQHG IRU PH ZLWK LPSODFDEOH VZHHWQHVV, LI HDFK GDB D IORZHU FOLPEV XS WR BRXU OLSV WR VHHN PH, DK PB ORYH, DK PB RZQ, LQ PH DOO WKDW ILUH LV UHSHDWHG, LQ PH QRWKLQJ LV HAWLQJXLVKHG RU IRUJRWWHQ, PB ORYH IHGGV RQ BRXU ORYH, EHORYHG, DQG DV ORQJ DV BRX OLYH LW ZLOO EH LQ BRXU DUPV ZLWKRXW OHDYLQJ PLQH.</div> <div>EB SDEOR QHUXGD</div>	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00. Accounting for previous tries, this gives **0.00/1.00**.

Question **3**

Correct

Mark 0.00 out of 1.00

Cho một danh sách sinh viên được lưu trữ trong một file văn bản có cấu trúc như bên dưới:

```
3
Nguyen Van A
5.6 7.9 10
Tran Van B
4 5 6
Bui Thi Tuong Van
7 9 4
```

- Dòng đầu tiên là một số nguyên n ($0 < n < 100$), mô tả số lượng sinh viên.
- $2n$ dòng tiếp theo lưu thông tin của các sinh viên. Mỗi sinh viên được lưu trong 2 dòng: dòng đầu là họ và tên (tối đa 50 ký tự), dòng thứ hai chứa 3 số thực mô tả điểm của 3 môn (Toán, Vật lý, Hoá học). Điểm nằm trong khoảng từ 0 đến 10.

Viết chương trình đọc danh sách các sinh viên từ một file văn bản theo định dạng như trên và in ra màn hình theo mẫu (thứ tự ngược lại với thứ tự trong file):

```
Toan |Vat ly |Hoa hoc |Ho va ten
7.00| 9.00 | 4.00 |Bui Thi Tuong Van
4.00| 5.00 | 6.00 |Tran Van B
5.60| 7.90 |10.00 |Nguyen Van A
Tong so: 3
```

Đầu vào

- Một chuỗi **s** mô tả tên file danh sách (s có nhiều nhất 100 ký tự)

Đầu ra

- In danh sách sinh viên theo mẫu trên

Chú ý

- Giá trị của các tham số luôn hợp lệ, không cần kiểm tra.
- Xem thêm chi tiết trong phần **For example**.

Gợi ý

- Khai báo một cấu trúc sinh viên để lưu trữ thông tin của một sinh viên
- Khai báo một mảng các sinh viên để lưu trữ các sinh viên
- Sử dụng chuỗi định dạng phù hợp để in điểm của các sinh viên.

For example:

Input	Result
file1.txt	Toan Vat ly Hoa hoc Ho va ten 7.00 9.00 4.00 Bui Thi Tuong Van 4.00 5.00 6.00 Tran Van B 5.60 7.90 10.00 Nguyen Van A Tong so: 3
file3.txt	Toan Vat ly Hoa hoc Ho va ten 9.50 10.00 8.50 Nguyen Truong Giang 6.90 4.50 10.00 Tran Huynh Ngoc Hien 6.50 7.50 9.80 Nguyen Vinh Tuong 10.00 3.50 9.50 Vo Phuoc Toan 7.90 7.50 4.50 Pham Minh Thang 9.50 5.50 6.80 Quach Huynh Huu Tai 8.90 8.50 10.00 Nguyen Thi Ngoc Nu Tong so: 7

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

```
1 #include <stdio.h>
2 #include <string.h>
3 typedef struct{
4     char ten[100];
5     float t,l,h;
6 }sv;
7 int main(){
8     FILE *f;
9     char name[10];
10    sv a[100];
11    int n,i;
```

```
12 fgets(name,10,stdin);
13 if(name[strlen(name)-1] == '\n'){
14     name[strlen(name)-1] = '\0';
15 }
16 f = fopen(name,"r");
17 fscanf(f,"%d\n", &n);
18 char h[100];
19 for(i = 0; i < n; i++){
20     fgets(h,100,f);
21     if(h[strlen(h)-1] == '\n'){
22         h[strlen(h)-1] = '\0';
23     }
24     strcpy(a[i].ten, h);
25     fscanf(f,"%f %f %f\n",&a[i].t, &a[i].l, &a[i].h);
26 }
27 fclose(f);
28 printf("Toan |Vat ly |Hoa hoc |Ho va ten\n");
29 for(i = n-1; i >= 0; i--){
30     printf("%5.2f|5.2f |5.2f |",a[i].t,a[i].l,a[i].h);
31     puts(a[i].ten);
32 }
33 printf("Tong so: %d",n);
34 return 0;
35 }
36 }
```

	Input	Expected	Got	
✓	file1.txt	Toan Vat ly Hoa hoc Ho va ten 7.00 9.00 4.00 Bui Thi Tuong Van 4.00 5.00 6.00 Tran Van B 5.60 7.90 10.00 Nguyen Van A Tong so: 3	Toan Vat ly Hoa hoc Ho va ten 7.00 9.00 4.00 Bui Thi Tuong Van 4.00 5.00 6.00 Tran Van B 5.60 7.90 10.00 Nguyen Van A Tong so: 3	✓
✓	file2.txt	Toan Vat ly Hoa hoc Ho va ten 6.90 4.50 10.00 Nguyen Thi Kim Ngan 6.50 7.50 9.80 Nguyen Gia Khoi 10.00 3.50 9.50 Nguyen Hoang Thanh Toan 7.90 7.50 4.50 Ho Viet Hung 4.50 5.50 6.80 Nguyen Minh Viet 8.60 8.50 10.00 Le Quoc Khanh Tong so: 6	Toan Vat ly Hoa hoc Ho va ten 6.90 4.50 10.00 Nguyen Thi Kim Ngan 6.50 7.50 9.80 Nguyen Gia Khoi 10.00 3.50 9.50 Nguyen Hoang Thanh Toan 7.90 7.50 4.50 Ho Viet Hung 4.50 5.50 6.80 Nguyen Minh Viet 8.60 8.50 10.00 Le Quoc Khanh Tong so: 6	✓
✓	file3.txt	Toan Vat ly Hoa hoc Ho va ten 9.50 10.00 8.50 Nguyen Truong Giang 6.90 4.50 10.00 Tran Huynh Ngoc Hien 6.50 7.50 9.80 Nguyen Vinh Tuong 10.00 3.50 9.50 Vo Phuoc Toan 7.90 7.50 4.50 Pham Minh Thang 9.50 5.50 6.80 Quach Huynh Huu Tai 8.90 8.50 10.00 Nguyen Thi Ngoc Nu Tong so: 7	Toan Vat ly Hoa hoc Ho va ten 9.50 10.00 8.50 Nguyen Truong Giang 6.90 4.50 10.00 Tran Huynh Ngoc Hien 6.50 7.50 9.80 Nguyen Vinh Tuong 10.00 3.50 9.50 Vo Phuoc Toan 7.90 7.50 4.50 Pham Minh Thang 9.50 5.50 6.80 Quach Huynh Huu Tai 8.90 8.50 10.00 Nguyen Thi Ngoc Nu Tong so: 7	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00. Accounting for previous tries, this gives 0.00/1.00.

Question **4**

Correct

Mark 0.33 out of 1.00

Mảng đối xứng gọi là PALINDROME. Ví dụ mảng 1.0 2.0 3.0 2.0 1.0 là PALINDROME.

Cho một mảng các số thực được lưu trữ trong tập tin văn bản như sau:

- Dòng đầu tiên ghi 1 số nguyên n là số phần tử của mảng
- Dòng thứ 2 ghi n số thực, mỗi số cách nhau khoảng trắng

Viết [hàm](#) checkPalindrome() để kiểm tra mảng số thực trong tập tin đầu vào có đối xứng hay không? Nếu đối xứng ghi lên tập tin đầu ra YES, ngược lại ghi NO

Khuôn dạng (Prototype)

- Tên [hàm](#): checkPalindrome
- Tham số:
 - char *sInput: tên file đầu vào
 - char *sOutput: tên file kết quả
- Kiểu trả về: void

Thân [hàm](#)

- Đọc nội dung file sInput, lưu trong mảng A.
- Kiểm tra mảng A đối xứng hay không? Nếu đối xứng ghi lên tập tin sOutput YES, ngược lại ghi NO

Gợi ý

- Dùng fscanf để đọc từ tập tin vào.
- Dữ liệu vào là hợp lệ, sinh viên không cần kiểm tra

Ví dụ file nguồn

```
5
1.0 2.0 3.0 2.0 1.0
```

In ra:

```
YES
```

For example:

Input	Result
palindrome1.txt	YES
palindrome2.txt	YES

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

```
1 int dx(float a[], int n){
2     int i;
3     for(i = 0; i < n/2-1; i++){
4         if(a[i] != a[n-1-i]){
5             return 0;
6         }
7     }
8     return 1;
9 }
10 void checkPalindrome(char *sInput, char *sOutput){
11     FILE *f;
12     int n;
13     float a[100];
14     int i;
15     f = fopen(sInput,"r");
16     fscanf(f,"%d", &n);
17     for(i = 0; i < n; i++){
18         fscanf(f,"%f", &a[i]);
19     }
20     fclose(f);
21     f =fopen(sOutput,"w");
22     if(dx(a,n)){
23         puts("YES");
24     }
25     else
26         puts("NO");
27     fclose(f);
28
29 }
```


	Input	Expected	Got	
✓	palindrome1.txt	YES	YES	✓
✓	palindrome2.txt	YES	YES	✓
✓	palindrome3.txt	YES	YES	✓
✓	palindrome4.txt	NO	NO	✓
✓	palindrome5.txt	NO	NO	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00. Accounting for previous tries, this gives **0.33/1.00**.

Question **5**

Correct

Mark 1.00 out of 1.00

Viết chương trình đọc một file văn bản và in nội dung của file ra màn hình

Đầu vào

- một chuỗi s - tên file (không chứa khoảng trắng). Tên file có ít hơn 10 ký tự. Mỗi dòng của file chứa ít hơn 100 ký tự

Đầu ra

- Nội dung file s

Chú ý

- Dữ liệu đầu vào luôn hợp lệ, không cần kiểm tra.
- Xem thêm chi tiết trong phần **For example**.

Gợi ý

- Mỗi dòng đều kết thúc bằng ký tự xuống dòng ('\n')
- Dòng cuối cùng có thể không chứa ký tự xuống dòng
- Đọc từng dòng cho đến khi không còn gì để đọc (tham khảo [hàm](#) fgets)

For example:

Input	Result
file1.txt	dong 1
	dong 3

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

```
1 #include <stdio.h>
2 #include <string.h>
3 int main(){
4     char s[10],str[100];
5     fgets(s,100,stdin);
6     if(s[strlen(s)-1] == '\n'){
7         s[strlen(s)-1] = '\0';
8     }
9     FILE *f;
10    f = fopen(s,"r");
11    while(fgets(str,100,f) != NULL){
12        if(str[strlen(str)-1] == '\n'){
13            str[strlen(str)-1] = '\0';
14        }
15        puts(str);
16    }
17    fclose(f);
18    return 0;
19 }
```

	Input	Expected	Got	
✓	file1.txt	dong 1	dong 1	✓
		dong 3	dong 3	
✓	file2.txt	dong 1	dong 1	✓
		dong 3	dong 3	
		dong 4	dong 4	

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question **6**

Correct

Mark 1.00 out of 1.00

Viết chương trình đọc một file văn bản và in các dòng không rỗng ra màn hình.

Đầu vào

- một chuỗi s - tên file (không chứa khoảng trắng). Tên file có ít hơn 10 ký tự. Mỗi dòng của file chứa ít hơn 100 ký tự

Đầu ra

- Nội dung file s

Chú ý

- Dữ liệu đầu vào luôn hợp lệ, không cần kiểm tra.
- Xem thêm chi tiết trong phần **For example**.

Gợi ý

- Mỗi dòng đều kết thúc bằng ký tự xuống dòng ('\n')
- Dòng cuối cùng có thể không chứa ký tự xuống dòng
- Đọc từng dòng cho đến khi không còn gì để đọc (tham khảo [hàm](#) fgets)

For example:

Input	Result
file1.txt	dong 1 dong 3

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

```
1 #include <stdio.h>
2 #include <string.h>
3 int main(){
4     char s[10],str[100];
5     fgets(s,100,stdin);
6     if(s[strlen(s)-1] == '\n'){
7         s[strlen(s)-1] = '\0';
8     }
9     FILE *f;
10    f = fopen(s,"r");
11    while(fgets(str,100,f) != NULL){
12        if(str[strlen(str)-1] == '\n'){
13            str[strlen(str)-1] = '\0';
14        }
15        if(strlen(str) > 1)
16            puts(str);
17    }
18    fclose(f);
19    return 0;
20 }
```

	Input	Expected	Got	
✓	file1.txt	dong 1 dong 3	dong 1 dong 3	✓
✓	file2.txt	dong 1 dong 3 dong 4	dong 1 dong 3 dong 4	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question **7**

Correct

Mark 1.00 out of 1.00

Viết [hàm](#) write_text để ghi giá trị của hai biến a và b vào file f.

Khuôn dạng (Prototype)

- Tên [hàm](#) (function name): **write_text**
- Tham số (parameters):
 - FILE *f**
 - int a**
 - int b**
- Kiểu trả về (return type): **void**

Thân hàm (Body)

- Lưu nội dung của biến a và biến b vào file f dưới dạng văn bản, hai số cách nhau 1 khoảng trắng. File f đã được mở cho phép ghi dưới dạng văn bản.

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**.

For example:

Test	Result
write_text(f, 4, 7);	4 7
write_text(f, -1, -4);	-1 -4

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

1

2

3

```
void write_text(FILE *f, int a, int b){  
    fprintf(f,"%d %d",a,b);  
}
```

	Test	Expected	Got	
✓	write_text(f, 4, 7);	4 7	4 7	✓
✓	write_text(f, 0, 70);	0 70	0 70	✓
✓	write_text(f, -1, -4);	-1 -4	-1 -4	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question **8**

Correct

Mark 0.00 out of 1.00

Cho một bảng hình chữ nhật kích thước $m \times n$ (m dòng, n cột), các dòng được đánh thứ tự từ trên xuống, các cột được đánh thứ tự từ trái sang. Có một ô trong bảng không nhận giá trị (ô trống), các ô còn lại nhận 1 giá trị từ $1..m \times n - 1$ (không có 2 ô có giá trị giống nhau).

Người chơi sẽ tiến hành hoán đổi ô trống và các ô có giá trị lân cận để đạt đến trạng thái kết thúc của trò chơi.

Trò chơi gọi là kết thúc nếu các giá trị $1..m \times n - 1$ lần lượt theo thứ tự được bố trí vào các ô $(1, 1), (1, 2), \dots, (1, n); (2, 1), (2, 2), \dots, (2, n), \dots; (m, 1), (m, 2), \dots, (m, n - 1)$.

Cho kích thước của bảng chữ nhật $m \times n$ và giá trị hiện hành của các ô trong bảng. Kiểm tra xem trò chơi đạt trạng thái kết thúc hay chưa?

Bảng chữ nhật được lưu trữ trong tập tin văn bản như sau:

- Dòng đầu tiên ghi 2 số nguyên m,n - mỗi số cách nhau khoảng trắng
- m dòng tiếp theo mỗi dòng ghi n giá trị (từ $0..m \times n - 1$), mỗi giá trị cách nhau 1 khoảng trắng. Giá trị 0 tại ô (i, j) cho biết ô (i,j) là ô trống.

Ví dụ:

```
3 3
1 2 3

6 5 4

7 8 0
```

Viết [hàm](#) isFinished() để kiểm tra bảng hình chữ nhật trong tập tin đầu vào biểu diễn cho trạng thái kết thúc hay không? Nếu có ghi lên tập tin đầu ra YES, ngược lại ghi NO

Khuôn dạng (Prototype)

- Tên [hàm](#): checkFinished
- Tham số:
 - char *sInput: tên file đầu vào
- Kiểu trả về: void

Thân [hàm](#)

- Đọc nội dung file sInput, lưu trong bảng A.
- Kiểm tra bảng A có biểu diễn trạng thái kết thúc của trò chơi Puzzle hay không? Nếu có hiển thị YES, ngược lại ghi NO

Gợi ý

- Dùng fscanf để đọc từ tập tin vào.
- Dữ liệu vào là hợp lệ, sinh viên không cần kiểm tra

Ví dụ file nguồn

Ví dụ 1:

```
3 3
1 2 3

6 5 4

7 8 0
```

--> NO

Ví dụ 2:

```
2 3
1 2 3

4 5 0
```

--> YES

For example:

Input	Result
puzzle1.txt	NO

Input	Result
puzzle2.txt	YES

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

```
1 typedef struct{
2     int m,n;
3     int A[100][100];
4 }mang;
5 int isFinished(mang a){
6     int i,j,flag = 1;
7     for (i = 0; i < a.m; i++){
8         for (j = 0; j < a.n; j++){
9             if(a.A[i][j] == 0){
10                 a.A[i][j] = a.m*a.n;
11             }
12             if (a.A[i][j] != (i*a.n+j+1))
13                 flag = 0;
14         }
15     }
16     return flag;
17 }
18 void checkFinished(char *sInput){
19     FILE *f;
20     int i,j;
21     mang s;
22     f = fopen(sInput,"r");
23     fscanf(f,"%d %d", &s.m, &s.n);
24     for(i = 0; i < s.m; i++){
25         for(j = 0; j < s.n; j++){
26             fscanf(f,"%d", &s.A[i][j]);
27         }
28     }
29     fclose(f);
30     if(isFinished(s)){
31         printf("YES");
32     }
33     else
34         printf("NO");
35 }
```

	Input	Expected	Got	
✓	puzzle1.txt	NO	NO	✓
✓	puzzle2.txt	YES	YES	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00. Accounting for previous tries, this gives **0.00/1.00**.

Question **9**

Correct

Mark 0.67 out of 1.00

Viết chương trình đọc một file văn bản và đếm xem file đó có bao nhiêu dòng.

Đầu vào

- một chuỗi s - tên file (không chứa khoảng trắng). Tên file có ít hơn 10 ký tự. Mỗi dòng của file chứa ít hơn 100 ký tự

Đầu ra

- Một số nguyên duy nhất là số dòng trong file

Chú ý

- Dữ liệu đầu vào luôn hợp lệ, không cần kiểm tra.
- Xem thêm chi tiết trong phần **For example**.

Gợi ý

- Mỗi dòng đều kết thúc bằng ký tự xuống dòng ('\n')
- Dòng cuối cùng có thể không chứa ký tự xuống dòng
- Đọc từng dòng cho đến khi không còn gì để đọc (tham khảo [hàm](#) fgets)

For example:

Input	Result
file1.txt	3

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

```
1 #include <stdio.h>
2 #include <string.h>
3 int main(){
4     char name[10],str[100];
5     FILE *f;
6     int count = 0;
7     fgets(name,10,stdin);
8     if(name[strlen(name)-1] == '\n'){
9         name[strlen(name)-1] = '\0';
10    }
11    f = fopen(name,"r");
12    while(fgets(str,100,f) != NULL){
13        count++;
14    }
15    fclose(f);
16    printf("%d",count);
17    return 0;
18 }
```

	Input	Expected	Got	
✓	file1.txt	3	3	✓
✓	file2.txt	4	4	✓

Passed all tests! ✓

Correct

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

Question **10**

Correct

Mark 1.00 out of 1.00

Viết chương trình đọc một file văn bản chứa một file nguồn của ngôn ngữ C. Hãy in nội dung của file này ra màn hình và bỏ qua các phần chú thích có trong file. Giả sử file nguồn chỉ chứa các chú thích dạng:

```
//chú thích
```

Ví dụ file nguồn

```
int main () { //khai bao ham main
    int a; //khai bao bien
    //lenh return
    return 0;
}
```

In ra:

```
int main () {
    int a;

    return 0;
}
```

Đầu vào

- một chuỗi s - tên file (không chứa khoảng trắng). Tên file có ít hơn 10 ký tự. Mỗi dòng của file chứa ít hơn 100 ký tự

Đầu ra

- Nội dung file s

Chú ý

- Dữ liệu đầu vào luôn hợp lệ, không cần kiểm tra.
- Xem thêm chi tiết trong phần **For example**.

Gợi ý

- Mỗi dòng đều kết thúc bằng ký tự xuống dòng ('\n')
- Dòng cuối cùng có thể không chứa ký tự xuống dòng
- Đọc từng dòng cho đến khi không còn gì để đọc (tham khảo [hàm](#) fgets), tìm vị trí của chuỗi \\ trong dòng (tham khảo [hàm](#) strstr() và phép toán trừ 2 con trỏ cho nhau)

For example:

Input	Result
source.c	<pre>int main () { int a; return 0; }</pre>

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

```
1  #include <stdio.h>
2  #include <string.h>
3  int main(){
4      FILE *f;
5      char name[10],str[100];
6      fgets(name,10,stdin);
7      if(name[strlen(name)-1] == '\n'){
8          name[strlen(name)-1] = '\0';
9      }
10     int i;
11     f = fopen(name,"r");
12     while(fgets(str,100,f) != NULL){
13         for(i = 0; i < strlen(str); i++){
14             if(str[i] == '/'){
15                 if(str[i] == '/'){
16                     printf("\n");
17                     break;
18                 }
19             }
20             printf("%c",str[i]);
21         }
22     }
23     fclose(f);
24     return 0;
```


25 | }

	Input	Expected	Got	
✓	source.c	int main () { int a; return 0; }	int main () { int a; return 0; }	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question **11**

Correct

Mark 0.33 out of 1.00

Mật mã Caesar là một phương pháp mã hoá cổ điển được đặt tên theo tên của Hoàng đế vĩ đại Julius Caesar. Tương truyền, Julius Caesar dùng phương pháp này để viết thư cho các người tình của mình.

Phương pháp mã hoá khá đơn giản: thay thế ký tự đang xét bằng ký tự đứng sau nó 3 ký tự trong bảng chữ cái, ví dụ: A được thay bằng D, B được thay bằng E, ..., X được thay bằng A, Y được thay bằng B, Z được thay bằng C.

Câu: THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG

Sẽ được mã hoá thành: WKH TXLFN EURZQ IRA MXPSV RYHU WKH ODCB GRJ

Để giải mã ta sẽ làm ngược lại: thay thế ký tự đang xét bằng ký tự đứng trước nó 3 ký tự trong bảng chữ cái:

- D -> A
- E -> B
- ...
- A -> X
- B -> Y
- C -> Z

Viết [hàm decrypt](#) đọc 1 file văn bản, mã hoá nội dung của nó theo phương pháp mã hoá Caesar và ghi nội dung đã mã hoá vào một file văn bản khác.

Khuôn dạng (Prototype)

- Tên [hàm](#): decrypt
- Tham số:
 - char *sInput: tên file cần mã hoá
 - char *sOutput: tên file kết quả
- Kiểu trả về: void

Thân hàm

- Đọc nội dung file sInput, mã hoá và ghi kết quả vào file sOutput.
- Quy tắc mã hoá như sau:
 - Chỉ giải mã các chữ cái (A - Z, a - z), các ký tự khác giữ nguyên
 - Đổi các chữ cái thành chữ hoa và giải mã chữ cái theo phương pháp mật mã Caesar.

Gợi ý

- Đọc từng dòng hoặc đọc từ ký tự, mã hoá, ghi kết quả ra file.
- Với từng ký tự là chữ cái A - Z chuyển nó về số 0 - 25, lấy số tương ứng cộng 23 ($23 = -3 + 26$) và chia cho 26 lấy phần dư (để đảm bảo các số nằm trong khoảng từ 0 - 25), chuyển số trở thành chữ cái lại A - Z.

For example:

Input	Result
file3.txt	THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG

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

```
1 void decrypt(char *sInput, char *sOutput){
2     FILE *f,*fn;
3     char s[100];
4     int i;
5     f = fopen(sInput,"r");
6     fn = fopen(sOutput,"w");
7     while(fgets(s,100,f) != NULL){
8         if(s[strlen(s)-1] == '\n'){
9             s[strlen(s)-1] = '\0';
10        }
11        for(i = 0; i < strlen(s); i++){
12            if(s[i] >= 'A' && s[i] <= 'Z'){
13                if(s[i] <= 'C'){
14                    s[i]+=26;
15                }
16                s[i]-=3;
17            }
18        }
19        fputs(s,fn);
20        fprintf(fn,"\n");
21    }
22    fclose(f);
23    fclose(fn);
24 }
```

	Input	Expected	Got	
✓	file3.txt	THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG	THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG	✓
✓	file4.txt	IF YOU FORGET ME: I WANT YOU TO KNOW ONE THING. YOU KNOW HOW THIS IS: IF I LOOK AT THE CRYSTAL MOON, AT THE RED BRANCH OF THE SLOW AUTUMN AT MY WINDOW, IF I TOUCH NEAR THE FIRE THE IMPALPABLE ASH OR THE WRINKLED BODY OF THE LOG, EVERYTHING CARRIES ME TO YOU, AS IF EVERYTHING THAT EXISTS, AROMAS, LIGHT, METALS, WERE LITTLE BOATS THAT SAIL TOWARD THOSE ISLES OF YOURS THAT WAIT FOR ME. WELL, NOW, IF LITTLE BY LITTLE YOU STOP LOVING ME I SHALL STOP LOVING YOU LITTLE BY LITTLE. IF SUDDENLY YOU FORGET ME DO NOT LOOK FOR ME, FOR I SHALL ALREADY HAVE FORGOTTEN YOU. IF YOU THINK IT LONG AND MAD, THE WIND OF BANNERS THAT PASSES THROUGH MY LIFE, AND YOU DECIDE TO LEAVE ME AT THE SHORE OF THE HEART WHERE I HAVE ROOTS, REMEMBER THAT ON THAT DAY, AT THAT HOUR, I SHALL LIFT MY ARMS AND MY ROOTS WILL SET OFF TO SEEK ANOTHER LAND. BUT IF EACH DAY, EACH HOUR, YOU FEEL THAT YOU ARE DESTINED FOR ME WITH IMPLACABLE SWEETNESS, IF EACH DAY A FLOWER CLIMBS UP TO YOUR LIPS TO SEEK ME, AH MY LOVE, AH MY OWN, IN ME ALL THAT FIRE IS REPEATED, IN ME NOTHING IS EXTINGUISHED OR FORGOTTEN, MY LOVE FEEDS ON YOUR LOVE, BELOVED, AND AS LONG AS YOU LIVE IT WILL BE IN YOUR ARMS WITHOUT LEAVING MINE. BY PABLO NERUDA	IF YOU FORGET ME: I WANT YOU TO KNOW ONE THING. YOU KNOW HOW THIS IS: IF I LOOK AT THE CRYSTAL MOON, AT THE RED BRANCH OF THE SLOW AUTUMN AT MY WINDOW, IF I TOUCH NEAR THE FIRE THE IMPALPABLE ASH OR THE WRINKLED BODY OF THE LOG, EVERYTHING CARRIES ME TO YOU, AS IF EVERYTHING THAT EXISTS, AROMAS, LIGHT, METALS, WERE LITTLE BOATS THAT SAIL TOWARD THOSE ISLES OF YOURS THAT WAIT FOR ME. WELL, NOW, IF LITTLE BY LITTLE YOU STOP LOVING ME I SHALL STOP LOVING YOU LITTLE BY LITTLE. IF SUDDENLY YOU FORGET ME DO NOT LOOK FOR ME, FOR I SHALL ALREADY HAVE FORGOTTEN YOU. IF YOU THINK IT LONG AND MAD, THE WIND OF BANNERS THAT PASSES THROUGH MY LIFE, AND YOU DECIDE TO LEAVE ME AT THE SHORE OF THE HEART WHERE I HAVE ROOTS, REMEMBER THAT ON THAT DAY, AT THAT HOUR, I SHALL LIFT MY ARMS AND MY ROOTS WILL SET OFF TO SEEK ANOTHER LAND. BUT IF EACH DAY, EACH HOUR, YOU FEEL THAT YOU ARE DESTINED FOR ME WITH IMPLACABLE SWEETNESS, IF EACH DAY A FLOWER CLIMBS UP TO YOUR LIPS TO SEEK ME, AH MY LOVE, AH MY OWN, IN ME ALL THAT FIRE IS REPEATED, IN ME NOTHING IS EXTINGUISHED OR FORGOTTEN, MY LOVE FEEDS ON YOUR LOVE, BELOVED, AND AS LONG AS YOU LIVE IT WILL BE IN YOUR ARMS WITHOUT LEAVING MINE. BY PABLO NERUDA	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00. Accounting for previous tries, this gives **0.33/1.00**.

