

مرتب سازی سریع

- محدودیت زمان: ۲ ثانیه
- محدودیت حافظه: ۱۲۸ مگابایت

مرتب سازی سریع از جمله روش های محبوب و با سرعت بالای برای مرتب کردن داده ها محسوب می شود. این روش هم مثل روش ادغامی از الگوریتم تقسیم و حل برای مرتب کردن داده ها استفاده می کند. به این ترتیب که داده ها را به دو قسمت مجزا تقسیم و با مرتب کردن آن ها کل داده ها را مرتب می کند. برای اینکار یکی از داده ها (مثلا داده اول) به عنوان محور انتخاب می شود. داده ها براساس محور طوری چینش می شوند که همه داده های کوچکتر از محور، سمت چپ و داده های بزرگتر یا مساوی با آن در سمت راستش قرار می گیرند. با مرتب کردن دو قسمت به دست آمده کل داده ها مرتب می شوند. در این حالت مثل روش ادغام نیازی به ادغام کردن داده ها نیست، چرا که قسمت سمت راست همگی از قسمت سمت چپ کوچکتر هستند و بالعکس. مثلاً اعداد صحیح زیر را در نظر بگیرید:

5 6 9 0 4 5 15 3 1 14 10

عدد 5 را به عنوان محور در نظر می گیریم. داده ها به این صورت بازچینی می شوند:

1 0 4 3 1 4 5 6 9 5 15 10

همانطور که مشاهده می کنید اعداد سمت چپ عدد 5 همگی از 5 کوچکتر اعداد سمت راست همگی بزرگتر یا مساوی آن هستند. در تصویر زیر چگونگی انجام این الگوریتم نشان داده شده است. با این تفاوت که محور الگوریتم، عنصر وسط در نظر گرفته شده است:



در تنها خط ورودی یک رشته از اعداد به شما داده می‌شود. اعداد از ۱۰۰۰۰۰۰۰۰۰ کوچکترند.

همچنین تعداد اعداد از ۵۰۰۰۰۰ کمتر است.

در تنها خط خروجی اعداد را به صورت مرتب شده صعودی چاپ کنید.

نمونه ورودی

0 1 5 3 6 2 4

0 1 2 3 4 5 6

ماشین متنی

- محدودیت زمان: ۱ ثانیه
- محدودیت حافظه: ۱۲۸ مگابایت

در این برنامه از شما خواسته شده است تا یک ماشین متنی کوچک با مجموعه‌ای از قابلیت های ساده را پیاده‌سازی نمایید. روال کار بدین صورت است که در آغاز کار برنامه یک رشته متنی اولیه (به طول حداکثر 1000) را از ورودی دریافت می‌کند. در ادامه تا زمانی که دستور خروج را دریافت کند در هر نوبت کاربر درخواست یک عملیات بر روی رشته متنی می‌دهد.

ورودی

عملیات ها به شرح زیر تعریف شده‌اند:

توضیحات و خروجی	دستور ورودی
SHIFT-R N	تمام کاراکترهای عبارت را به صورت چرخشی N واحد به سمت راست منتقل می‌کند.
SHIFT-L N	تمام کاراکترهای عبارت را به صورت چرخشی N واحد به سمت چپ منتقل می‌کند.
EXTEND N	به انتهای رشته موجود N کاراکتر جدید اضافه می‌کند و به عنوان مقدار پیشفرض کاراکترها، ستاره (*) قرار می‌دهد.
SHRINK N	از انتهای رشته، N کاراکتر حذف می‌کند. در صورتی که طول رشته کمتر از N بود، رشته حاصل یک رشته خالی خواهد بود.
REVERSE	رشته را معکوس می‌کند.
PUT I C	حرف مکان Iام رشته را با حروف C جایگزین می‌کند. توجه داشته باشید که شماره مکان‌ها از یک آغاز می‌شود و I همواره کوچکتر مساوی طول رشته خواهد بود.

توضیحات و خروجی	دستور ورودی
PRINT	رشته فعلی را چاپ می‌کند و به خط بعد می‌رود.
EXIT	اتمام برنامه

خروجی

برنامه شما تنها به ازای عملیات چاپ خروجی خواهد داشت و به ازای سایر دستورات صرفاً عملیات موردنظر را بر روی رشته متنی انجام می‌دهد. در پیاده‌سازی این سوال، شما باید به ازای تمامی دستورات (به جز خروج) یک تابع در نظر بگیرید و انجام عملیات توسط فراخوانی آن تابع انجام بگیرد. به عنوان نمونه امضای توابع باید به این صورت باشد:

```
void Extend( char *string, int _extendedLength);
```

مثال

ورودی نمونه ۱

```
initial string
PRINT
EXTEND 2
SHIFT-R 3
PRINT
PUT 3 0
REVERSE
SHRINK 2
PRINT
EXIT
```

خروجی نمونه ۱

```
initial string
g**initial strin
```

nirts laitinio

ورودی نمونه ۲

Test
PRINT
SHRINK 20
PRINT
EXTEND 2
PRINT
EXIT

خروجی نمونه ۲

Test

**

مرتب‌سازی رشته‌ها

- محدودیت زمان: ۱ ثانیه
- محدودیت حافظه: ۵۰ مگابایت

در این برنامه شما باید به تعداد نامعلومی رشته از کاربر بگیرید. روش دریافت ورودی به این شکل است که کاربر شروع به وارد کردن کلمه‌های موردنظر خود می‌کند. این عملیات تا جایی ادامه پیدا می‌کند که کاربر 0 را وارد کند. برنامه باید تمام کلماتی را که تا اینجا وارد شده است، به ترتیب حروف الفبای انگلیسی چاپ کند.

تذکر:

- حروف وارد شده تنها شامل حروف کوچک و بزرگ انگلیسی هستند.
- حروف بزرگ به حروف کوچک اولویت دارند.
- برای اطلاعات بیشتر درباره‌ی نحوه‌ی ذخیره‌سازی کاراکترها، عبارت ASCII Table را گوگل کنید.
- کلمات با کاراکتر کمتر، به کلمات با کاراکتر بیشتر اولویت دارند؛ به عنوان مثال، a زودتر از aa چاپ می‌شود.
- در مرتب‌سازی کلمات، تنها دو حرف اول آن نگاه می‌کنیم؛ بنابراین اگر دو کلمه در حرف اول مشترک بودند، به سراغ حرف دوم می‌رویم. اگر در حرف دوم هم یکسان بودند، هر دو کلمه از نظر مرتب‌سازی هم‌ارزش‌اند؛ بنابراین کلمه‌ای که در ورودی زودتر وارد شده است، زودتر چاپ می‌شود.
- استفاده از توابع آماده مرتب‌سازی رشته‌ها در این سوال، قابل قبول نیست و کد مرتب‌سازی به طور کامل باید توسط شما نوشته شده باشد.

ورودی

در خط اول ورودی تعدادی کلمه می‌آید که با صفر خاتمه می‌یابد. (تعداد و مجموع طول کلمات کمتر از ۱۰۰۰ است.)

خروجی

در خروجی تمامی کلمات داده شده را به ترتیب حروف الفبای انگلیسی چاپ کنید.

مثال

نمونه ورودی:

alireza Mohammad Arash anahita sarah Milad john Alireza Maryam 0

نمونه خروجی :

Alireza Arash alireza anahita john Maryan Milad Mohammad sarah

کمی پوینتر

تکه کد پیوست شده حاوی خطاهایی در استفاده از اشاره‌گرهاست که مانع از کارکرد صحیح آن شده و ممکن است باعث بروز خطا به هنگام اجرای برنامه شوند. کد زیر را با حداقل تغییر اصلاح کرده و بارگذاری کنید. اصلاحات انجام شده روی کد را کامنت کنید.

تابع main() فاقد هرگونه اشکال نوشته شده است تا Prototype توابع را بدانید. تغییری در آن ایجاد نکنید. همچنین تمام کتابخانه‌های قابل استفاده برای حل سوال Include شده‌اند و اجازه اضافه کردن به آن‌ها را ندارید.

کد شما دستی نیز بررسی میشود (:)

###نکته: اشاره‌گرها را با عملگرهایی همچون [] جایگزین نکنید.

#کد :

```
#include <stdio.h>
#define SIZE 100

void read(int *array, int size) {
    int i;
    for (i = 0; i < size; i++)
        scanf("%d", *array + i);
}

void swap(int *a, int *b) {
    int *temp = a;
    a = b;
    b = temp;
}

void print(const int *array[], int size) {
    int i;
    for (i = 0; i < size; i++)
        printf("%d ", **(array + i));
}

void sort(int array[], const int size) {
    int *end = array + size, i;
```



```

        for (i = array; i != end - 1; i++)
            if (i > (i + 1))
                swap(i, i + 1);
    }

    int isSorted(const int array[], const int *size) {
        int i;
        for (i = 0; i < *size - 1; i++)
            if (*(array + i) > *(array + i + 1))
                return 0;
        return 1;
    }

    int main() {
        int arr[SIZE], len;
        scanf("%d", &len);
        read(arr, len);
        sort(arr, &len);
        printf("%s\n", isSorted(arr, &len) ? "All is good!" : "This isn't going
        print(arr, len);
        return 0;
    }

```

ورودی نمونه

نداریم

خروجی نمونه

نداریم

****ورودی خروجیامون تموم شدن برین از مغازه بقلی بگیرین (=**