Михайленко Ярослав [204 группа, 3 семестр, 2022-2023]: Submit a solution

Problem mz06-1-mz06-1

Full score:	100
Run penalty:	10
Time limit:	1 s
Real time limit:	5 s
Memory limit:	64M
Stack limit:	8M
Open date:	2022/10/10 10:30:00
Deadline:	2022/10/10 12:05:00

Problem mz06-1: mz06-1

В аргументах командной строки задается путь к каталогу.

На стандартный поток вывода напечатайте количество файлов в заданном каталоге (без подкаталогов), удовлетворяющих следующему условию:

- Файл регулярный.
- Файл доступен на выполнение текущему пользователю.
- Его имя заканчивается на суффикс `.exe`.

Символические ссылки прослеживайте до соответствующих файлов (используйте соответствующий системный вызов семейства *stat).

Не используйте системные вызовы, меняющие текущий каталог процесса.

View
all / necessary

up02-4
up02-5
up02-6
up02-7
mz06-1
mz08-1
mz08-2
mz08-3
mz08-4
mz08-5

Previous submissions of this problem

Run ID	Time	Size	Problem	Language	Result	Tests passed	Score	View source	View report
1768	2022/10/10 12:00:59	884	mz06-1	gcc-32	Pending review	4	**50**=100-6*10	View	View
1765	2022/10/10 11:58:53	897	mz06-1	gcc-32	Partial solution	0	**0**=0-5*10	View	View
1753	2022/10/10 11:45:51	822	mz06-1	gcc-32	Partial solution	0	**0**=0-4*10	View	View
1751	2022/10/10 11:45:30	822	mz06-1	gcc-32	Coding style violation	N/A	N/A	View	View
1747	2022/10/10 11:41:18	814	mz06-1	gcc-32	Partial solution	0	**0**=0-3*10	View	View
1744	2022/10/10 11:38:47	813	mz06-1	gcc-32	Partial solution	0	**0**=0-2*10	View	View
1740	2022/10/10 11:35:34	799	mz06-1	gcc-32	Partial solution	0	**0**=0-1*10	View	View
1724	2022/10/10 11:23:48	719	mz06-1	gcc-32	Partial solution	0	**0**	View	View
1723	2022/10/10 11:23:31	702	mz06-1	gcc-32	Coding style violation	N/A	N/A	View	View
1719	2022/10/10 11:20:19	702	mz06-1	gcc-32	Compilation error	N/A	N/A	View	View
1717	2022/10/10 11:19:50	702	mz06-1	gcc-32	Compilation error	N/A	N/A	View	View
1714	2022/10/10 11:18:00	694	mz06-1	gcc-32	Compilation error	N/A	N/A	View	View

Problem mz06-2-mz06-2

Full score:	100
Run penalty:	10
Time limit:	1 s
Real time limit:	5 s
Memory limit:	1G
Memory RSS limit:	512M
Stack limit:	8M
Open date:	2022/10/10 10:30:00
Deadline:	2022/10/10 12:05:00

Problem mz06-2: mz06-2

Напишите функцию:

```
void normalize_path(char *buf);
```

Функция принимает на вход строку абсолютного пути. Строка всегда начинается с символа '/', в строке не встречаются несколько символов '/' подряд.

Если входная строка не равна "/" и оканчивается на '/', то этот символ с конца должен быть удален.

В строке-пути должны быть удалены все вхождения каталогов `.` и `..` исходя из того, что каталог `.` — это переход на текущий каталог, а каталог `..` — это переход на родительский каталог.

Функция модифицирует область памяти, в которой находится входная строка. Функция должна обрабатывать пути произвольной длины, не обязательно ограниченной константой PATH_MAX.

Запрещается использовать стандартные функции.

Examples

Input

```
/a/b/c/
```

Output

```
/a/b/c
```

Input

```
/a/b/../
```

Output

```
/a
```

Input

```
/a/b/../../..
```

Output

```
/
```

View
all / necessary

up02-4
up02-5
up02-6
up02-7
mz06-2
mz08-1
mz08-2
mz08-3
mz08-4
mz08-5

Problem mz06-3-mz06-3

Full score:	100
Run penalty:	10
Time limit:	1 s
Real time limit:	5 s
Memory limit:	64M
Stack limit:	8M
Open date:	2022/10/10 10:30:00
Deadline:	2022/10/10 12:05:00

View
all / necessary

up02-4
up02-5
up02-6
up02-7
mz06-3
mz08-1
mz08-2
mz08-3
mz08-4
mz08-5

Problem mz06-3: mz06-3

Дезинфекция помещения выполняется каждый второй и четвертый четверг каждого месяца, кроме четвергов, номер дня в месяце которых делится на три. То есть, если четверг приходится на 3, 6, 9 и т. д. число месяца, дезинфекция не производится.

На стандартном потоке ввода задается год в диапазоне от 1910 до 2037 Григорианского календаря.

На стандартный поток вывода выведите все даты дезинфекции. Каждая дата уборки представляется двумя числами: номер месяца (1-12), день в месяце (1-31). Месяц и день разделяйте пробелом. Каждую дату выводите на отдельной строке.

Для работы с календарем используйте функции из time.h.

Examples

Input

```
2020
```

Output

```
1 23
2 13
3 26
4 23
5 14
5 28
6 11
6 25
7 23
8 13
9 10
10 8
10 22
11 26
12 10
```

Problem mz06-4-mz06-4

Full score:	100
Run penalty:	10
Time limit:	1 s
Real time limit:	5 s
Memory limit:	64M
Stack limit:	8M
Open date:	2022/10/10 10:30:00
Deadline:	2022/10/10 12:05:00

Problem mz06-4: mz06-4

В аргументе командной строки передается путь к каталогу D и максимальный размер Z.

На стандартный поток вывода напечатайте относительные пути к регулярным файлам, которые доступны на чтение текущему пользователю и имеют размер не больше Z. Подкаталоги обрабатывайте рекурсивно на глубину не более 4. Каждый путь выводите на отдельной строке. Относительные пути не должны содержать каталогов `.` и `..`

Символические ссылки игнорируйте.

Глубиной 1 считается каталог, заданный в аргументах командной строки. Относительный путь к файлу на глубине 4 запишется в виде `a/b/c/d` Это - максимальная глубина, которая должна обрабатываться.

Не используйте `scandir`, `ftw` и аналогичные функции. Не используйте `chdir`, `fchdir`, `telldir`, `seekdir`, а также `getpwd` и аналогичные. Можно использовать `opendir`, `readdir`, `closedir`, access и `*stat`.

Например.

```
a
b/c
d/e/f/g
```

View
all / necessary

up02-4
up02-5
up02-6
up02-7
mz06-4
mz08-1
mz08-2
mz08-3
mz08-4
mz08-5

Problem mz06-5-mz06-5

Full score:	100
Run penalty:	10
Time limit:	1 s
Real time limit:	5 s
Memory limit:	64M
Memory RSS limit:	1G
Stack limit:	8M
Open date:	2022/10/10 10:30:00
Deadline:	2022/10/10 12:05:00

Problem mz06-5: mz06-5

Напишите функцию:

```
ssize_t getcwd2(int fd, char *buf, size_t size);
```

Параметр `fd` — это открытый файловый дескриптор каталога, `buf` — адрес начала буфера в памяти, `size` — размер буфера.

Функция должна восстановить полный путь к каталогу, который открыт и связан с файловым дескриптором `fd`. То есть, если fd — корневой каталог, то полный путь к каталогу — "/". Если fd — не корневой каталог, то полный путь к этому каталогу начинается с "/", состоит из записей в соответствующих каталогах, являющихся каталогами, но не символическими ссылками. На конце полного пути "/" должен отсутствовать. Например, "/usr/local/bin" (без кавычек).

Если при выполнении функции не возникло ошибок при выполнении системных вызовов и библиотечных функций, функция `getcwd2` должна вернуть длину строки — полного пути к каталогу (без учета \0). В противном случае функция должна вернуть отрицательное значение.

Если параметр `size` равен 0, то функция должна вернуть только длину пути и больше не выполнять никаких действий (не модифицировать память по указателю `buf`). Если `size` больше нуля, то по адресу `buf` должен быть записан восстановленный полный путь к каталогу, но не более чем `size-1` значащих символов и байт \0. Байт \0 должен быть записан в любом случае.

Функция должна сохранять текущий рабочий каталог процесса, то есть текущий рабочий каталог процесса до вызова функции должен совпадать с текущим рабочим каталогом процесса после вызова функции. Файловый дескриптор с номером fd должен остаться открытым и связанным с тем же (исходным) каталогом.

Из функций и системных вызовов работы с файлами и файловой системой разрешается использовать только opendir, closedir, readdir, stat, lstat, fstat, chdir, fchdir, dirfd. Функции работы со строками и символами разрешается использовать без ограничений.

View
all / necessary

up02-4
up02-5
up02-6
up02-7
mz06-5
mz08-1
mz08-2
mz08-3
mz08-4
mz08-5