

# File

Nathan is playing around with his laptop. He is a tech-savvy person and likes to experiment with his laptop. One day, Nathan is bored and starts to play around with the files inside his laptop. However, he is not satisfied with the available file explorer system in his laptop. He is confident that he can design a much better file explorer than the one installed in his laptop.

Nathan wants to write his own console language software to manage the files in his laptop. This simple application should be able to do the following things:

1. Create files and folders
2. Delete a file from a folder
3. Move a file from a folder to another folder
4. Find the folder with the largest size

Nathan is an experienced programmer, but he is too busy with his work. Therefore, he wants you to help him develop this simple file manager. As a start, Nathan gives you this design as a guideline to follow:

File	
-	<b>name</b> : String
-	<b>size</b> : int
-	<b>folderName</b> : String

Each file has its own name and size. All files are located inside a folder.

Folder	
-	<b>name</b> : String
-	<b>listOfFiles</b> : List

Each folder has its own name. All folders may contain some files, but not another folder.

Nathan is quite happy with the design specification that he gave you. Your job is simple: help Nathan create the file explorer system he dreams of.

## Input

The first line of input consists of a single integer **Q** ( $1 \leq Q \leq 100$ ), the number of operations. **Q** rows follow. In each of the **Q** rows, there are queries that you have to answer. The queries will follow the following specification:

Query Type    Input Format: **<QUERY\_TYPE> <APPROPRIATE\_PARAMETERS>**

1.    **Createfile FILE\_NAME FILE\_SIZE FOLDER\_NAME**  
A file with name **FILE\_NAME** with size **FILE\_SIZE** is created inside the folder **FOLDER\_NAME**. It is guaranteed that the folder **FOLDER\_NAME** exists when this operation is called (i.e. it has been created beforehand) and there are no files with the name **FILE\_NAME** in the system before this operation is called.
2.    **Createfolder FOLDER\_NAME**  
A new folder with the name **FOLDER\_NAME** is created. It is guaranteed that there are no folders with the name **FOLDER\_NAME** before this operation is called.

3. **Deletefile FILE\_NAME**  
Delete file with the name **FILE\_NAME**. It is guaranteed that the file exists in the system when this operation is called.
4. **Count FOLDER\_NAME**  
Print the total size of all files located inside the folder **FOLDER\_NAME**. It is guaranteed that the folder with the name **FOLDER\_NAME** exists when this operation is called.
5. **Movefile FILE\_NAME FOLDER\_NAME**  
Move the file **FILE\_NAME** to the folder **FOLDER\_NAME**. It is guaranteed that the file with the name **FILE\_NAME** and the folder with the name **FOLDER\_NAME** exist when this operation is called.
6. **Findlargest**  
Print the name of the folder that has the largest size. It is guaranteed that there is only one largest folder in the system.

All folders and files have unique names and all operations are valid as stated in the specification above. All names are guaranteed to be at most 20 characters and consist of only lowercase letters and / or numbers. All file sizes are at most 2000.

### Output

Print according to the specification of the queries given above.

Sample Input	Sample Output
12	30
Createfolder lab	15
Createfile lab1 10 lab	25
Createfile lab2 20 lab	tutorial
Createfolder tutorial	lab
Createfile tutorial1 15 tutorial	
Count lab	
Count tutorial	
Movefile lab1 tutorial	
Count tutorial	
Findlargest	
Movefile tutorial1 lab	
Findlargest	