Coding Arena

Problem

More and more programming languages today are embracing the phenomenon of automatic memory management. As a Language Development Lead your supervisor is responsible for managing memory allocation routines of the Language Compiler. Since this is beta release of the language your supervisor is developing, he wants to keep memory management routines clean and simple.

Rules for allocating memory are quite simple :

1. Continuous memory locations can be merged together to form a “Free List”. Free Lists are maintained via indexes in simple data structures.
2. When it comes to memory allocations, routines should prefer Exact-Fit over First-Fit
3. If and only if there is no Exact-Fit, the routines should find the First-Fit where the object can be allocated.

You are developing this memory-subsystem as per specifications provided by your supervisor. Go ahead and crack the problem. Specifications are as depicted below :

Input Format:

Line 1	Input absolute path of the memory representation file
Line 2	Memory allocation required by the object

File Format:

Each memory representation file consists of 4 column separated by a comma. File will end with a New Line character. The 4 fields are as depicted below :

Index Id, Start Offset, File Size, Delete Flag

Output Format:

Print “Inserted at index I”,where I is position where insertion is made or “Can not be inserted”,if insertion in any index is not possible

Line 1	**For Valid Input,print** “Inserted at index I”,where I is position where insertion is made **Or** “Can not be inserted” if insertion is not possible

Sample Test Cases:

File:Input.txt

1,1,100,N

2,101,250,N

3,352,100,Y

4,453,50,N

5,504,100,Y

6,605,100,Y

7,606,100,N

SNo.	Input	Output	Explanation
1	Input.txt 200	Inserted at index 5	Object 5 and 6 are marked for deletion using Delete Flag as 'Y'. Hence these contiguous memory locations are merged to form a Free list with Index 5. Since the Free List is of size 200 and the Object is also of size 200, the routine should

			find this as an Exact-Fit.
2	Input.txt 80	Inserted at index 3	Object 3 is marked for deletion using Delete Flag as 'Y'. Since objects 2 and 4 are still Live, Free list with Index 3 is only size 100. Object is of size 80. Since there is no Exact-Fit the routine should find allocation for this object by First-Fit method.
3	Input.txt 210	Can not be inserted	

Note:

Participants submitting solutions in C language should not use functions from <conio.h> / <process.h> as these files do not exist in gcc

Submit Answer

☐ I BHARGAVA GANTI confirm that the answer submitted is my own.

Browse... Submit