A common method to store records in a file is the random-access file, where all the file is viewed as a byte stream and all the records has the same size. One of the advantage of random-access file is that it can directly manipulate any position in byte of the file. Your goal is to write a program to create and manipulate a random-access file.
**Requirement: Read the binary input from file stream rather than the standard input stream. Also write the binary output to the file stream rather than the standard output stream.**

## Input

The input is also a random-access file named as “**test_input_RandomAccess.txt**” which contains 50000 records. Each record includes a unsigned integer as the offset value, and a character as the byte value.

## Output

Create a 64 MB random-access file named as “**my_output_RandomAccess.txt**”. Use the input offset and byte value to manipulate the corresponding bytes in the random-access file. For example, for a record with input offset 30 and byte value 20, you should modify the 30th byte in random-access file by the value 20. Note that the input offset is started from 0.
(Hint: to create a large size of array, you should dynamically allocate the memory through the following expression:
*type* pointer = (type*) malloc(size);*
where *type* is the data type you need and *size* is the array size you need.
Remember to free the memory through
free(pointer);
when you no longer need to use the space, possibly before the end of your program.)

## Sample Input

No visible sample input.

## Sample Output

No visible sample output.