4. Write a menu driven program to implement the strict version of binomial min-heap; that is, after any operation, no two trees in the heap should have the same rank. Your program must implement the following functions.

MAKE-HEAP(): Create and return a new heap containing no elements.

INSERT(*H, x*): Insert node *x* into heap *H*.

MINIMUM(*H*): Return the value of the smallest key in the heap *H*. The function should run in O(1) time, in the worst case.

UNION(*H1, H2*): Create and return a new heap that contains all the nodes of heaps *H1* and *H2*. Heaps *H1* and *H2* are "destroyed" by this operation.

EXTRACT-MIN(*H*): Remove the node from heap *H* whose key is minimum and return a pointer to the node.

DECREASE-KEY(*H, x, k*): If node *x*'s key is at least *k*, decrease the value of its key by *k*. Otherwise, print -1.

DELETE(*H, x*): Delete node *x* from heap *H*.

Additionally, the program should maintain an array *A* (in the main function) to hold pointers to the nodes in the heap, as detailed in the **input format** section below.

Input Format

The first line of the input contains a single integer *n*, the size of the array *A*.

Each of the following lines contains a string from the set {'insr', 'min', 'extr', 'decr', 'del', 'tc', 'stop'}, followed by zero, one or two integers, as specified below.

`insr j k`	- If *A[j]* is not pointing to any node in the heap, then insert a new node with key 'k' into the heap and add a pointer from *A[j]* to the new node.
`min`	- Print the value of the smallest key in the heap.
`extr`	- Print the value of the smallest key in the heap and delete the node containing this key.
`decr j k`	- If *A[j]* is pointing to a node with key at least *k*, decrease the key by *k*.
`del j`	- Print the key of the node pointed to by *A[j]* and delete the node.
`tc`	- Print the number of trees currently in the heap.
`stop`	- Terminate the program.

Output Format

If an operation cannot be performed, print -1.

Sample Input and Output

Input:

```
10
insr 0 10
insr 2 20
insr 5 30
insr 1 40
insr 3 50
extr
insr 0 60
decr 3 45
extr
tc
min
del 5
```

Output:

```
10
5
1
20
30
```