```
#include <stdio.h>
#include <stdlib.h>

void moveToEnd(int *arrMplusN, int size)
{
	int index, right = size-1;
	for(index = size-1; index >=0; index--)
		if(arrMplusN[index])
			arrMplusN[right--] = arrMplusN[index];
}

void merge(int *arrMplusN, int *arrN, int sizeM, int sizeN)
{
	int indexM = sizeN, indexN = 0, final = 0;
	while(final < sizeM + sizeN)
	{
		if ((indexM < (sizeM + sizeN) && arrMplusN[indexM] <= arrN[indexN])
		 || (indexN == sizeN))
			arrMplusN[final++] = arrMplusN[indexM++];
		else
			arrMplusN[final++] = arrN[indexN++];
	}
}

void printMplusN(int *arrMplusN, int size)
{
	for(int index = 0; index < size; index++)
		printf("%d ", arrMplusN[index]);
}

int main()
{
	int sizeM, sizeN;
	// O Represents array index value is empty
	int arrMplusN[] = {3, 9, 0, 0, 14, 0, 16, 0, 25};
	int arrN[] = {6, 8, 10, 30};
	sizeN = sizeof(arrN) / sizeof(arrN[0]);
	sizeM = sizeof(arrMplusN)/ sizeof(arrMplusN[0]) - sizeN;

	//move the m elements to end of arrMplusN
	moveToEnd(arrMplusN, sizeM+sizeN);

	//merge two arrays
	merge(arrMplusN, arrN, sizeM, sizeN);

	printMplusN(arrMplusN, sizeM+sizeN);
	return 0;

}
```

Time complexity: O(m + n)
Space complexity: O(1)