|  |
| --- |
| #define ALLOCSIZE 10000 */\* size of available space \*/* static char array[ALLOCSIZE]; */\* storage for alloc \*/* static char \*allocp = array; */\* next free position \*/*   char \*alloc(int n){ */\* return pointer to n characters \*/*  if (array + ALLOCSIZE - allocp >= n) { */\* it fits \*/*  allocp += n;  return allocp - n; */\* old p \*/*  }  else */\* not enough room \*/*  return 0; } void afree(char \*start\_point , char \* end\_point) {*/\* free storage pointed to by p \*/*  if (start\_point >= array && start\_point < array + ALLOCSIZE && end\_point >= array && end\_point< array + ALLOCSIZE){  if(end\_point == allocp) { *// if deleting last allocation*  allocp = start\_point;  }  else{ *// otherwise*  while(end\_point != allocp){    end\_point++;  \*start\_point = \*end\_point;  start\_point++; /\* tiny edited here \*/  }  allocp = start\_point;  }  }  } |
|  |