

.MODEL FLAT	
PUBLIC towers ;	; esporta towers
EXTERN printf:NEAR	; importa printf
.CODE	
-towers: PUSH EBP	; salva EBP (puntatore al record d'attivazione) e decrementa ESP
MOV EBP, ESP	; imposta nuovo puntatore al record al di sopra di ESP
CMP [EBP+8], 1	; if ( n == 1 )
JNE L1	; salta se n è diverso da 1
MOV EAX, [EBP+16]	; printf ( " .. " , i, j );
PUSH EAX	; si noti che i parametri i, j e la stringa
MOV EAX, [EBP+12]	; di formattazione sono impilati sullo stack
PUSH EAX	; in ordine inverso. È la convenzione per le chiamate C
PUSH OFFSET FLAT:format	; è l'indirizzo della stringa formato
CALL _printf	; chiama printf
ADD ESP, 12	; rimuove i parametri dallo stack
JMP Done	; abbiamo concluso
L1 : MOV EAX, 6	; comincia k = 6 - i - j
SUB EAX, [EBP+12]	; EAX = 6 - i
SUB EAX, [EBP+16]	; EAX = 6 - i - j
MOV [EBP+20], EAX	; k = EAX
PUSH EAX	; comincia towers(n - 1 , i, k)
MOV EAX, [EBP+12]	; EAX = i
PUSH EAX	; push di i
MOV EAX, [EBP+8]	; EAX = n
DEC EAX	; EAX = n - 1
PUSH EAX	; push di n - 1
CALL _towers	; chiama towers(n - 1, i, 6 - i - j)
ADD ESP, 12	; rimuove i parametri dallo stack
MOV EAX, [EBP+16]	; comincia towers(1, i, j)
PUSH EAX	; push di j
MOV EAX, [EBP+12]	; EAX = i
PUSH EAX	; push di i
PUSH 1	; push di 1
CALL _towers	; chiama towers(1, i, j)
ADD ESP, 12	; rimuove i parametri dallo stack
MOV EAX, [EBP+12]	; comincia towers(n - 1 , 6 - i - j, j)
PUSH EAX	; push di j
MOV EAX, [EBP+20]	; EAX=k
PUSH EAX	; push di k
MOV EAX, [EBP+8]	; EAX = n
DEC EAX	; EAX = n - 1
PUSH EAX	; push di n - 1
CALL _towers	; chiama towers(n - 1, 6 - i - j, j)
ADD ESP, 12	; modifica il puntatore allo stack
Done: LEAVE	; preparazione all'uscita
RET 0	; ritorno al chiamante
.DATA	
format DB 'Move disk from %d to %d\n'	; stringa di formattazione
END	