

1) E' dato un programma che, letto in input un numero intero n, stampa tutti i numeri binari con n cifre.

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

int potenza(int a, int b) {
    int i, x=1;
    for (i=1; i<=b; i++) {
        x = x*a; }
    return(x); }

void stampa_numeri(int n) {
    int i, j, x;
    for (j=0; j<potenza(2,n); j++) {
        x=j;
        for (i=n-1; i>=0; i--) {
            if (x==0)
                printf("0");
            else
                if (x>=potenza(2,i) {
                    printf("1");
                    x = x-potenza(2,i); }
                else { printf("0"); }
        }
        printf("\n"); }
    return; }
```

```
int main(void) {
    int n;
    printf("Inserisci n: ");
    scanf("%d", &n);
    stampa_numeri(n);
    return(0); }
```

a) Calcolare la complessità computazionale  $T(n)$  dell'algoritmo in numero di passi base, con n e m dimensioni dell'input (*motivare brevemente le scelte e il metodo applicato durante il calcolo*);

b) Calcolare la complessità asintotica dell'algoritmo (Procedere ri-calcolando la complessità di ogni macroistruzione, applicando opportunamente le regole per blocchi di istruzioni in sequenza o annidati.)

2) Consideriamo una sequenza di caratteri A e B variabile con  $n > 0$  numero di A ed  $m > 0$  numero di B scritti sul nastro iniziale. Si costruisca una Macchina di Turing che termini con un nastro finale contenente la sola lettera A se  $n > m$ , con la sola lettera B se  $n < m$  o con la sola lettera C se  $n = m$ . Si descriva a parole, prima della matrice funzionale, l'algoritmo di cui la matrice sarà descrizione.