

Nome: _____ Cognome: _____ Matr: _____ Postazione _____

Scrivere un programma in linguaggio C (chiamare il progetto con la propria <matricola>) che abbia il comportamento descritto nel seguito. Il tempo a disposizione è di **2 ore**. Al termine del tempo gli elaborati salvati su **U:** verranno raccolti automaticamente dal sistema. Eventuali documenti sono disponibili in **T:\Bertozzi**, usare **wordpad** per leggere i file di testo.

Le “Parole di Dyck” possono esser viste come stringhe formate da $2n$ caratteri ($x_1x_2x_3x_4\dots x_{2n-1}x_{2n}$) di due soli tipi (ad esempio ‘0’ e ‘1’). Assumendo di usare una funzione di valutazione $h()$ tale per cui

$$h(0)=1, \quad h(1)=-1 \quad \text{e} \quad h(x_1x_2x_3\dots x_k)=\sum_{i=1}^k h(x_i), \quad \text{un stringa } x_1x_2x_3\dots x_{2n} \text{ è una parola}$$

di Dyck se e solo se $h(x_1x_2x_3\dots x_{2n})=0$ (il numero di simboli ‘1’ è pari al numero di simboli ‘0’) e $h(x_1x_2x_3\dots x_i) \geq 0$ per $1 \leq i \leq 2n-1$ (una sottostringa iniziale qualunque della parola in esame contiene un numero di ‘0’ superiore o uguale al numero di ‘1’).

Si vuole individuare tutte le rappresentazioni binarie (di $2n$ bit) di numeri interi non negativi che sono anche Parole di Dyck, a tal fine si sviluppi un programma in C che:

1. Chieda all’utente un numero di bit e verifichi che il numero inserito sia pari
2. Generi tutti i numeri interi (senza segno) rappresentabili in formato binario con il numero di bit inserito
3. Verifichi se la rappresentazione binaria dei numeri generati sia anche una Parola di Dyck. A tal fine si sviluppino e si usino due funzioni:
 - a) **unsigned int bit(unsigned long x, unsigned int i, unsigned int nbits)** che restituisce il bit i -esimo della rappresentazione binaria del numero x assumendo che $i=1$ corrisponda al bit più significativo dei $nbits$ bit da analizzare
 - b) **int h(unsigned long x, unsigned int i, unsigned int nbits)** che restituisce il valore della funzione di valutazione $h()$, precedentemente descritta, dei primi i bit **-più significativi-** del numero x
4. Quando la verifica effettuata al punto precedente ha esito positivo si stampi il numero e la sua rappresentazione binaria usando il numero di bit inserito. Si stampi inoltre il numero totale di parole di Dyck individuate

Esempio di output:

Inserisci un numero di bit da analizzare (pari): 6

```
7: 000111
11: 001011
13: 001101
19: 010011
21: 010101
```

Il numero totale di parole di Dyck trovate e' 5

Il codice va sviluppato nell’ordine indicato. Per ogni punto implementato si verifichi il corretto funzionamento. La correzione termina al primo punto non implementato correttamente. Il programma deve poter gestire rappresentazioni binarie fino a 48 bit. L’implementazione del punto 3.b in maniera ricorsiva permette di ottenere un punteggio aggiuntivo.