

Laboratorio Informatica A - Incontro 2

Problema 1: Array

Scrivere un programma che acquisisca una sequenza di 10 numeri interi ed un indice X tra 0 e 9. Il programma dovrà stampare la somma dei numeri in posizioni minori di X e il prodotto dei numeri in posizioni successive a X.

Problema 2: Voti

Scrivere un programma che richiede l'inserimento di voti ottenuti in un certo numero di corsi. Il programma chiede qual è il numero di voti da inserire. Per ciascuno, l'utente deve inserire numero di crediti e voto ottenuto (in 30imi). Il programma infine stampa la media pesata dei corsi e il numero totale dei crediti. Si ricorda che la media pesata, ad esempio su due voti, si calcola come:

$$\text{media} = (\text{voto1} * \text{crediti1} + \text{voto2} * \text{crediti2}) / (\text{crediti1} + \text{crediti2}).$$

Problema 3: Divisori

Scrivere un programma che chiede due numeri all'utente e ne stampa a video tutti i divisori comuni.

Problema 4: Numeri Primi

Si scriva un programma in grado di decidere se un numero dato dall'utente è primo. Si ricorda che un numero è primo se non è divisibile per nessuno dei valori compresi tra il numero stesso e 1.

Problema 5: Sequenza

Scrivere un programma che acquisisca due sequenze A e B di 5 numeri interi ciascuna, e stampi a video la sequenza "interlacciata" tra A e l'inversa di B: il primo elemento di A, l'ultimo elemento di B, il secondo elemento di A, il penultimo di B etc. Esempio di esecuzione:

Dammi i 5 valori di A: > 1 2 3 4 5

Dammi i 5 valori di B: > 6 7 8 9 10

Stampa: 1 10 2 9 3 8 4 7 5 6

Problema 6: Scomposizione

Scrivere un programma che chiede un numero all'utente e ne stampa a video la scomposizione in fattori primi.

Esempi: $27=3^3$ $12=2^2*3$ $36=2^2*3^2$ $120=2^3*3*5$

Problema 7: Guess a number

Scrivere un programma che generi un numero intero casuale e chieda all'utente di indovinarlo. Ad ogni tentativo fallito da parte dell'utente di indovinare il numero il programma indica se il numero da indovinare è maggiore o minore rispetto a quello appena inserito.

Per generare un numero casuale è necessario chiamare una volta la funzione `srand(time(0))` per inizializzare il generatore di numeri casuali ed in seguito, per ogni numero casuale, la funzione `rand()` come di seguito.

Es. `r = rand()`

N.B. Si noti che la funzione `rand()` restituisce un numero casuale tra zero e la costante `RAND_MAX` (costante molto grande). E' possibile usare il solito operatore `%` per limitare questo range (L'uso della funzione `time(..)` necessita della libreria `time.h`. Le funzioni `srand(..)` e `rand(..)` sono invece parte di `stdlib.h`).