

Laboratorio Informatica A

Incontro 1

Problema 1: Tabelline

Scrivere un programma per il calcolo delle tabelline. Il programma riceve dall'utente due numeri. Il primo corrisponde al numero di cui è necessario calcolare la tabellina mentre il secondo indica la lunghezza della tabellina richiesta.

Es: 2, 10 produce come risultato: 2 4 6 8 10 12 14 16 18 20

Es: 3, 8 produce come risultato: 3 6 9 12 15 18 21 24

Accettare solamente numeri positivi: se l'utente inserisce un numero negativo mostrare un messaggio di errore e richiede un altro numero.

Problema 2: Divisori

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

Problema 3: Calcolatrice

Scrivere un programma che permette di effettuare le operazioni di una semplice calcolatrice: *Somma*, *Sottrazione*, *Divisione*, *Prodotto*. Il programma deve quindi richiedere all'utente la tipologia di operazione da svolgere (es. "premere 1 se si vuole effettuare una somma" ...) ed in seguito i due operandi necessari. Dopo ogni operazione stampare i risultati ottenuti. **NB:** Attenzione alla divisione!

Estensione:

Aggiungere anche l'elevamento a potenza (calcolato moltiplicando il numero per sé stesso) come operazione disponibile.

Problema 4: 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. **N.B.** 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`).

Problema 5: Google

Alla fine degli anni '90 due squattrinati studenti di Stanford (Larry Page e Sergey Brin) vi propongono di investire su una loro idea: un innovativo motore di ricerca per Internet. Voi siete alquanto scettici sulla possibile riuscita del progetto e volete valutare i possibili ritorni economici. Scrivete quindi un programma che riceve in ingresso 3 valori (usare dei `float`): il capitale investito, una percentuale di interesse ed il numero di anni su cui valutare l'investimento. Per ogni anno aggiornare gli interessi:

$$\text{interessi} = \text{capitale} * \text{percentuale_interessi} / 100$$

ed aggiornare il capitale sommando gli interessi ottenuti:

$$\text{capitale} = \text{capitale} + \text{interessi}$$

Stampare a video per ogni anno il valore del capitale aggiornato.

Problema 6: 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. **N.B.** Per la verifica della condizione di divisibilità si può utilizzare l'operatore `%` come nel problema 1.

Estensione:

Aggiungere anche il calcolo del fattoriale del numero inserito

Problema 7: Fibonacci

Scrivere un programma che chieda un intero positivo `n` all'utente e in risposta stampi i primi `n` numeri della serie di Fibonacci. La serie di Fibonacci è definita come segue:

$$F(0) = 1 \qquad F(1) = 1$$

$$F(n) = F(n-1) + F(n-2)$$

Quindi la serie è:

1 1 2 3 5 8 13