

15 OTTOBRE 2019 – Incontro 3 di 8 – C base

Esercizio 1

Scrivere un programma che chieda continuamente un numero intero positivo n all'utente fino a quando viene inserito il valore 0. Il programma deve stampare sullo schermo tanti asterischi '*' quanti sono indicati dal numero n .

Infine il programma deve stampare a video quanti asterischi sono stati stampati in totale.

```
numero di '*'? 5
```

* * * * *

numero di '*'? 3

* * *

numero di '*'? 0

```
stampati 8 '*'
```

Esercizio 2

La tavola pitagorica è una matrice bidimensionale i cui elementi sono numeri interi calcolati moltiplicando l'indice della riga con l'indice della colonna alle quali l'elemento appartiene.

Scrivere un programma che stampi a video una tavola pitagorica di dimensioni 10 x 10.

Gli elementi della matrice che risultano essere multipli sia di 5 che di 7 (contemporaneamente, ad esempio 35, 70, ...) devono essere evidenziati affiancandoli con un asterisco (*).

NOTE:

- non utilizzare i vettori o le matrici (stampare direttamente a video gli elementi)
- definire la dimensione 10 con una costante DIM_PIT: `#define DIM_PIT 10`
- non è obbligatorio allineare esattamente i numeri in colonna e inserire gli indici di riga e colonna durante la stampa, ma se si desidera farlo si ricorda che per stampare un numero intero `i` occupando ad esempio 4 “spazi” la sintassi è: `printf("%4d", i);`

Tavola Pitagorica 10 x 10:

	1	2	3	4	5	6	7	8	9	10
1	1	2	3	4	5	6	7	8	9	10
2	2	4	6	8	10	12	14	16	18	20
3	3	6	9	12	15	18	21	24	27	30
4	4	8	12	16	20	24	28	32	36	40
5	5	10	15	20	25	30	35*	40	45	50
6	6	12	18	24	30	36	42	48	54	60
7	7	14	21	28	35*	42	49	56	63	70*
8	8	16	24	32	40	48	56	64	72	80
9	9	18	27	36	45	54	63	72	81	90
10	10	20	30	40	50	60	70*	80	90	100

Esercizio 3

Data una costante N definita uguale a 7, scrivere un programma che visualizzi una figura quadrata composta da N righe e N colonne come quella indicata:

*+++++
 -*++++
 --*+++
 ---*+++
 ----*++
 -----*+
 -----*

Lungo la diagonale principale compaiono degli asterischi '*', al di sotto della diagonale compaiono dei segni meno '-' e al di sopra compaiono dei segni più '+'.

Esercizio 4

Scrivere un programma che generi una serie di numeri interi pseudo-casuali da 0 a 99.

I numeri vanno continuamente generati fino a quando si verifica almeno una delle seguenti condizioni che terminano la serie:

1. la quantità totale di numeri interi generati supera 10 e la quantità di numeri pari è maggiore di almeno due unità rispetto alla quantità di numeri dispari.
2. la quantità totale di numeri interi generati raggiunge 20.

NB: il testo specifica la condizione per far terminare il ciclo di generazione e quindi occorre “negarla” per avere la corrispondente condizione per far continuare il ciclo.

Per verificare che il programma funzioni correttamente, si consiglia di far stampare al programma quanti numeri sono stati generati e inoltre quanti di questi sono pari e quanti sono dispari.

Per utilizzare il generatore di numeri pseudo-casuali occorre:

- includere le seguenti librerie:

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

- inizializzare all'inizio del programma il generatore di numeri in base all'istante di tempo corrente (quanti secondi sono trascorsi dal 1 gennaio 1970) altrimenti ad ogni esecuzione del programma verrà generata sempre la stessa serie di numeri. Inserire questa istruzione fuori dal ciclo di generazione dei numeri altrimenti verrà generato sempre lo stesso numero.

```
//inizializza il generatore di numeri pseudo-casuali
srand(time(NULL));
```

- richiedere al generatore un numero tramite l'istruzione `rand()` e calcolarne il resto della divisione intera per ottenere un numero nell'intervallo desiderato

```
//genera un numero pseudo-casuale da 0 a 99
num = rand() % 100;
```

Esempio in cui il programma termina per la condizione 1):

Generato 44, in totale generato 1 numero pari

...

Generato 36, in totale generati 16 numeri di cui 9 pari e 7 dispari

La serie e' terminata perche' sono stati generati almeno 10 numeri e i pari superano di almeno 2 i dispari.

Esempio di esecuzione in cui il programma termina per la condizione 2):

Generato 79, in totale generato 1 numero dispari

...

Generato 71, in totale generati 20 numeri di cui 9 pari e 11 dispari

La serie e' terminata perche' sono stati generati 20 numeri.

Esercizio 5

Data la costante SUPERIORE definita uguale a 10, scrivere un programma che per ogni numero X compreso nell'intervallo 1..SUPERIORE (estremi compresi) stampi la somma dei numeri tra i 7 numeri successivi a X che hanno la stessa parità di X (ovvero se X è pari occorre sommare solo i numeri pari, se X è dispari occorre sommare solo i numeri dispari)

X= 1 dispari somma = 3 + 5 + 7 = 15

X= 2 pari somma = 4 + 6 + 8 = 18

X= 3 dispari somma = 5 + 7 + 9 = 21

X= 4 pari somma = 6 + 8 + 10 = 24

...

X=10 pari somma = 12 + 14 + 16 = 42

Esercizio 6

Scrivere un programma che chieda all'utente un numero intero $\text{num} \geq 0$ e stampi a video le sue cifre in lettere. Ad esempio, se $\text{num} = 9876$, il risultato sarà "nove otto sette sei".

Con un primo ciclo calcolare la prima potenza di $10 > n$ (es $10000 > 9876$). L'esponente da dare a 10 per ottenere tale potenza rappresenta il numero di cifre di cui è composto il numero (es 4 per 9876) e guiderà il numero di iterazioni del ciclo descritto di seguito.

In un altro ciclo occorre dividere num per potenza ottenendo così la cifra da stampare in lettere (con `printf` in un costrutto `switch`) e all'iterazione successiva ripetere l'operazione sul resto della divisione intera di num per potenza usando una potenza più piccola.

```
num (>=0)? 9876
Cerco la prima potenza di 10 > 9876...
10 <= 9876... 100 <= 9876... 1000 <= 9876...
10000 > 9876 -> trovato esponente = 4
Stampo le cifre dividendo per le potenze di 10 e tenendo il resto...
9876 / 1000 = 9 con resto = 876 -> "nove"
876 / 100 = 8 con resto = 76 -> "otto"
76 / 10 = 7 con resto = 6 -> "sette"
6 / 1 = 6 con resto = 0 -> "sei"
```

Esercizio 7

Scrivere un programma che simuli con un grafico il moto rettilineo uniformemente accelerato di un oggetto nel vuoto. Per la simulazione occorre impostare i seguenti parametri iniziali:

- la velocità iniziale dell'oggetto in metri al secondo **v0_ms** (1.0 m/s ovvero 3.6 km/h)
- l'accelerazione costante in metri al secondo² **a_ms2** (0.1 m/s²)
- la distanza totale che l'oggetto deve percorrere in metri **s_tot_m** (100 m ovvero 0.1 km)
- la lunghezza del grafico di simulazione in numero di caratteri **LEN_CHAR** (50 caratteri)
- l'intervallo in secondi **T_STEP_S** che trascorre a ogni "passo" della simulazione (10 s)

La formula per calcolare la velocità (**v_ms** in metri/secondo) ad un certo istante (**t_s** in secondi) data la velocità iniziale (**v0_ms** in metri/secondo) e l'accelerazione costante (**a_ms2** in metri/secondo²) è la seguente:

```
v_ms = v0_ms + a_ms2 * t_s;
```

La formula per calcolare la distanza percorsa (**s_m** in metri) ad un certo istante (**t_s** in secondi) data l'accelerazione (**a_ms2** in metri/secondo²) e la velocità iniziale (**v0_ms** in metri/secondo) è la seguente:

```
s_m = 1.0 / 2.0 * a_ms2 * (t_s * t_s) + v0_ms * t_s;
```

La formula della proporzione per calcolare la distanza (**s_char** in numero di caratteri) dell'oggetto su una riga lunga **LEN_CHAR** caratteri data la distanza percorsa (**s_m** in metri) per stampare a video la riga che rappresenta il "grafico" della simulazione è la seguente:

```
s_char = s_m * (LEN_CHAR / s_tot_m);
```

Per stampare il "grafico" occorre quindi stampare una riga di **LEN_CHAR** caratteri (in posizione **s_char** sarà presente il carattere 'x' che rappresenta il punto in cui si trova l'oggetto mentre nelle altre posizioni verrà stampato il carattere '-').

La simulazione termina quando l'oggetto percorre tutta la distanza ovvero quando **s_m > s_tot_m**.

NB: velocità in km/h = velocità in m/s * 3.6.

```
INIZIO SIMULAZIONE v0_ms=1.00 a_ms2=0.10 s_tot_m=100.00 step_s=10.0
Tempo = 0.0 s Velocita' = 1.000 m/s = 3.60 km/h Distanza = 0.00 m = 0.000 km
0 X-----100.00 m
Tempo = 10.0 s Velocita' = 2.000 m/s = 7.20 km/h Distanza = 15.00 m = 0.015 km
0 -----X-----100.00 m
Tempo = 20.0 s Velocita' = 3.000 m/s = 10.80 km/h Distanza = 40.00 m = 0.040 km
0 -----X-----100.00 m
Tempo = 30.0 s Velocita' = 4.000 m/s = 14.40 km/h Distanza = 75.00 m = 0.075 km
0 -----X-----100.00 m
Tempo = 40.0 s Velocita' = 5.000 m/s = 18.00 km/h Distanza = 120.00 m = 0.120 km
0 -----100.00 m
FINE SIMULAZIONE
```