

LABORATORIO DI FONDAMENTI DI INFORMATICA

18 OTTOBRE 2024 - Incontro 2 di 5 – C Base

Esercizio 1

Scrivere un programma che richieda all'utente una stagione tramite l'inserimento di un carattere ['e' per estate o 'i' per inverno] e una temperatura in gradi Centigradi.

Il programma deve stampare a video un messaggio per indicare se la temperatura è compresa o non è compresa nella media stagionale, considerando i seguenti intervalli di temperature:

- 'e'estate: temperatura nella media se compresa tra 27.5 e 32.5 °C (estremi inclusi)
 - 'i'nverno: temperatura nella media se compresa tra 2.5 e 7.5 °C (estremi inclusi)
- Stagione? e
Temperatura? 15.5
Temperatura estiva NON nella media stagionale!

Suggerimenti:

- Si consiglia di utilizzare una variabile `stagione` di tipo char che assumerà valori 'e' oppure 'i' (i valori di tipo char nel codice C si indicano inserendoli tra apici singoli).
- Si consiglia di separare bene i casi “innestando” una dentro l'altra le condizioni di if:

```
if("è estate") {
    if("T nella media estiva") {...}
    else {...}
} else if("è inverno") {
    if("T media invernale") {...}
    else {...}
}
else {... stagione non prevista ...}
```
- La condizione “T nella media” non può essere scritta come `if(27.5 <= T <= 32.5)` perché in questo modo non fornisce il risultato atteso, ma va scritta diversamente...

Esercizio 2

Scrivere un programma che chieda all'utente un numero intero num e stampi a schermo “vero” quando

(il valore di num è dispari) oppure NON (è compreso tra i valori 20 e 90 inclusi).

Se ciò non si verifica il programma deve stampare “falso”.

num?

5 -> vero 18 -> vero 20 -> falso 40 -> falso

Suggerimenti:

- Testare il corretto funzionamento del programma fornendo in input dei valori di `num` che verifichino i vari casi (ad es. num pari e compreso nell'intervallo; num pari e non compreso nell'intervallo; num = 90; num = 20; ...)
- L'operatore di negazione in C è il punto esclamativo !
- L'espressione logica può essere scritta in vari modi ricordando i Teoremi di De Morgan:

NOT (A AND B)	equivale a	NOT A OR NOT B
NOT (A OR B)	equivale a	NOT A AND NOT B

Esempio: `!(x >= 0 || x == -3)` equivale a `x < 0 && x != -3`
- Attenzione (`num < 20 && num > 90`) non un'espressione corretta perché è sempre falsa dato che un numero non può essere contemporaneamente `< 20` e `> 90`.

Esercizio 3

Scrivere un programma che simuli una semplice calcolatrice. Il programma deve richiedere all'utente due valori interi x e y e un carattere op . Tale carattere determina l'operazione da eseguire tra x e y ('+' per somma, '-' per differenza, '*' per moltiplicazione, '/' per divisione, 'M' per max, 'm' per min). Il risultato dell'operazione va stampato a video riportando l'operazione eseguita (ad es. " $x + y = <\text{risultato}>$ "). Se l'utente inserisce un carattere diverso da quelli elencati stampare un messaggio d'errore.

```
x? 5  
y? 3  
operazione? ...  
5 + 3 = 8      5 - 3 = 2      5 * 3 = 15      5 / 3 = 1.666667  
MAX(5, 3) = 5  min(5, 3) = 3  Operazione non valida
```

Suggerimenti:

- Utilizzare un costrutto `switch` per gestire la scelta dell'operazione desiderata
- Ricordare di inserire `fflush(stdin)` o `getchar()` o `scanf("%c", &op)` per svuotare il buffer dal carattere di invio prima di chiedere all'utente l'operazione op
- Per eseguire un cast da un tipo all'altro, ad esempio per trasformare un intero in un numero float per fare in modo che la divisione dia un risultato decimale, il nuovo tipo va inserito tra parentesi tonde. Es: `(float) x`

Esercizio 4

Scrivere un programma che dopo aver richiesto all'utente un numero intero continua a dividere successivamente per 2 tale numero visualizzando ogni volta il resto della divisione intera fin quando la divisione fornisce come risultato 0. (Se letti alla rovescia i resti danno la rappresentazione binaria del numero)

Numero da convertire in binario: 29
29 / 2 = 14 con resto di 1
14 / 2 = 7 con resto di 0
7 / 2 = 3 con resto di 1
3 / 2 = 1 con resto di 1
1 / 2 = 0 con resto di 1



Suggerimenti:

- Si consiglia di utilizzare 3 variabili intere: n , $resto$, $risultato$
- Implementare un ciclo `while` (oppure `do .. while`). Ad ogni iterazione n viene diviso per due stampando $resto$. Il ciclo termina quando la divisione di n per 2 restituisce come risultato 0.

Esercizio 5

Scrivere un programma che acquisiti due numeri interi positivi a e b , con $b \geq a$, calcoli e visualizzi il conteggio dei numeri nell'intervallo (estremi inclusi) e la loro somma e media aritmetica.

```
a? 4  
b? 12  
intervallo [4, 12]: n = 9; somma = 72; media = 8.000
```

Suggerimenti:

- Implementare un ciclo `for` in cui l'indice parte da a , l'indice aumenta con incrementi di 1 e termina quando l'indice supera b
- Media aritmetica = somma / numero_elementi

Esercizio 6

Scrivere un programma che calcoli e stampi a video i valori delle temperature da -20°C a 197°C (incluse), con incrementi di 7°C mostrando il corrispondente valore convertito in gradi Fahrenheit. Evidenziare nel messaggio stampato la temperatura (in gradi Centigradi) che produce una differenza tra i valori in gradi Fahrenheit e in gradi Centigradi ($^{\circ}\text{F} - ^{\circ}\text{C}$) pari a 100.

```
-20.00 C -> -4.00 F  
-13.00 C -> 8.60 F  
-6.00 C -> 21.20 F  
... Differenza 185.00 F - 85.00 C = 100 ...  
197.00 C -> 386.60 F
```

Suggerimenti:

- Implementare un ciclo `for` il cui indice parte da -20, incrementa a passi di 7 e fa terminare il ciclo una volta raggiunto 197
- Formula di conversione: $\text{F} = 32 + (\text{C} * 9.0 / 5.0)$
- Le temperatura da evidenziare è 85°C ($185^{\circ}\text{F} - 85^{\circ}\text{C} = 100$)

Esercizio 7

Scrivere un programma che continui a richiedere numeri interi (anche negativi) all'utente fino a quando il numero intero fornito è uguale a 0. Il programma deve stampare a video la quantità di numeri inseriti, la somma dei numeri inseriti e il maggiore e il minore tra quelli inseriti (sempre escludendo l'ultimo inserimento del numero zero).

Se viene inserito subito il numero 0, non si devono calcolare quantità, somma, min e max.

Numero? -5	Numero? 4	Numero? 0
Numero? 10	Numero? 0	Nessun calcolo da eseguire
Numero? 0	1 numero inserito	
2 numeri inseriti	somma = 4	
somma = 5	min = 4	
min = -5	max = 4	
max = 10		

Suggerimenti:

- Assegnare come valori iniziali delle variabili contenenti il minimo e il massimo il primo numero inserito dall'utente, andandole poi ad aggiornare se necessario ad ogni nuovo inserimento.
- Attenzione a non considerare 0 come minimo o massimo o parte del conteggio della quantità

Esercizio 8

Scrivere un programma che richieda all'utente un numero intero $0 < h < 10$. Il programma deve stampare consecutivamente tutti i numeri tra 1 e h ; deve poi stampare un triangolo rettangolo formato da h righe di caratteri asterisco '*'.

```
Altezza? 4  
1234  
*  
**  
***  
****
```

Suggerimenti:

- per disegnare il triangolo occorrono due cicli "innestati"
 - l'indice del ciclo "più esterno" sarà il numero di riga
 - l'indice del ciclo "più interno" sarà il numero di colonna.
- Il numero di colonne da stampare dipende dall'indice di riga, quindi la condizione di arresto del ciclo più interno dipende dall'indice del ciclo più esterno
- Occorre ricordare di "andare a capo" dopo aver stampato ciascuna riga

Esercizio 9

Scrivere un programma che prenda in considerazione tutte le coppie di valori interi (n_1, n_2) consecutivi (cioè $n_2 = n_1 + 1$) nell'intervallo 1..200 (estremi inclusi; n_1 si ferma a 199) e stampi a video le coppie di valori consecutivi n_1 e n_2 , quando questi sono entrambi divisibili per 3 o per 5 “indipendentemente” (cioè quando n_1 è divisibile per 3 **o** per 5 **e** n_2 è divisibile per 3 **o** per 5).

(5, 6) (9, 10) (20, 21) (24, 25) ... (189, 190)

Suggerimenti:

- è sufficiente un ciclo for con indice n_1 . n_2 si calcola come $n_1 + 1$
- un numero è divisibile per un altro se il resto della divisione intera tra i due è 0

Esercizio 10

Scrivere un programma che richieda all'utente 4 numeri interi: $a, b, x, n_{tentativi}$.

a e b (con $a > 0, a < b$) rappresentano gli estremi di un intervallo in cui cercare, tramite un algoritmo simile a quello della ricerca binaria, il numero x (con $a \leq x \leq b$), impiegando al massimo un numero di tentativi pari a $n_{tentativi} (> 0)$. Il programma deve procedere per passi successivi, dimezzando se necessario a ogni tentativo l'intervallo di ricerca in questo modo: prendendo la parte sinistra dell'intervallo, cioè ponendo $b=(a+b)/2$, nel caso in cui $x < (a+b)/2$, o la parte destra dell'intervallo, cioè ponendo $a=(a+b)/2$, altrimenti. Il programma, se la ricerca ha successo, cioè se $x=(a+b)/2$, entro il numero di tentativi prefissati, si ferma stampando il messaggio “trovato”; se invece il programma non dovesse riuscire a trovare x entro il numero di tentativi specificato deve stampare il messaggio “non trovato”.

Esempio di esecuzione con $a = 0, b = 100, x = 30, n_{tentativi} = 10$:

```
a? 0
b? 100
x? 30
numero massimo di tentativi? 10
Tentativo 1/10 Cerco 30 a=0 b=100 (a+b)/2=50 : non trovato - cerco a sx
Tentativo 2/10 Cerco 30 a=0 b=50 (a+b)/2=25 : non trovato - cerco a dx
Tentativo 3/10 Cerco 30 a=25 b=50 (a+b)/2=37 : non trovato - cerco a sx
Tentativo 4/10 Cerco 30 a=25 b=37 (a+b)/2=31 : non trovato - cerco a sx
Tentativo 5/10 Cerco 30 a=25 b=31 (a+b)/2=28 : non trovato - cerco a dx
Tentativo 6/10 Cerco 30 a=28 b=31 (a+b)/2=29 : non trovato - cerco a dx
Tentativo 7/10 Cerco 30 a=29 b=31 (a+b)/2=30 : trovato!
```