

LABORATORIO DI FONDAMENTI DI INFORMATICA

8 OTTOBRE 2019 – Incontro 2 di 8 – C base

Esercizio 1

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”. 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; ...)

`num?`

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

Esercizio 2

Scrivere un programma che chieda all'utente due valori interi da inserire nelle variabili `num1` e `num2`.

Calcolare il risultato (intero) della divisione e il resto (intero) della divisione di questi due numeri e inserirli rispettivamente nelle variabili `risultato` e `resto` per poi stampare a video il messaggio

`"num1 diviso num2 restituisce <risultato> con resto <resto>."`

Inoltre se `resto` vale zero indicare in un altro messaggio se `risultato` è pari o dispari.

`num1? 10`

`num2? 5`

`10 diviso 5 restituisce 2 con resto 0.`

`Resto zero -> risultato 2 pari.`

Esercizio 3

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 somma, min e max.

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.

`Numero? -5`

`Numero? 10`

`Numero? 0`

`2 numeri inseriti`

`somma = 5`

`min = -5`

`max = 10`

`Numero? 4`

`Numero? 0`

`1 numero inserito`

`somma = 4`

`min = 4`

`max = 4`

`Numero? 0`

`Nessun calcolo da eseguire`

Esercizio 4

Scrivere un programma che prenda in considerazione tutte le coppie di valori interi (`n1`, `n2`) consecutivi (cioè `n2` = `n1` + 1) nell'intervallo 1..200 (estremi inclusi; `n1` si ferma a 199) e stampi a video le coppie di valori consecutivi `n1` e `n2`, quando questi sono entrambi divisibili per 3 o per 5 “indipendentemente” (cioè quando `n1` è divisibile per 3 o per 5 e `n2` è divisibile per 3 o per 5).

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

Esercizio 5

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 (°F - °C) pari a 100.

NOTE: $F = 32 + (C * 9.0/5.0)$

Le temperatura da evidenziare è 85°C (185°F - 85°C = 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
```

Esercizio 6

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 = <risultato>"). Se l'utente inserisce un carattere diverso da quelli elencati stampare un messaggio d'errore.

NOTE:

- utilizzare un costrutto switch per gestire la scelta dell'operazione desiderata
- ricordarsi 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, il nuovo tipo va inserito tra parentesi tonde. Es: (float)

```
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
```

Esercizio 7

Scrivere un programma che richieda all'utente 4 numeri interi: a, b, x, ntentativi. 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 ntentativi (> 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 $a = 0$, $b = 100$, $x = 30$, $ntentativi = 10$:

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!