

Il prezzo della benzina è di euro 1.50 al litro. Scrivere una procedura che chiede all'utente quanti litri vuole comprare e stampa il prezzo corrispondente.

```
litri = raw_input("Quanti litri di benzina vuoi? ")
print "il costo è: ", float(litri) * 1.50
```

Scrivere una procedura che chiede all'utente tre numeri e stampa il loro prodotto.

```
x = raw_input("scrivi tre numeri")
y = raw_input()
z = raw_input()
print "il prodotto è: ", float(x)*float(y)*float(z)
```

Scrivere una procedura che svolge il seguente compito:

- chiede a un piccolo imprenditore il nome dei suoi tre dipendenti e il numero di ore che hanno lavorato durante il giorno;
- assumendo che la paga è di dieci euro l'ora, stampa il nome e la paga giornaliera di ognuno.

```
dip1 = raw_input("nome primo dipendente: ")
ore1 = raw_input("numero ore: ")
dip2 = raw_input("nome secondo dipendente: ")
ore2 = raw_input("numero ore: ")
dip3 = raw_input("nome terzo dipendente: ")
ore3 = raw_input("numero ore: ")
print dip1, 10*int(ore1), "euro"
print dip2, 10*int(ore2), "euro"
print dip3, 10*int(ore3), "euro"
```

Scrivere una procedura che chiede all'utente tre numeri e stampa la loro somma con una funzione.

```
def output(x,y,z):
    print "la somma è: ", x+y+z
x = raw_input("scrivi tre numeri")
y = raw_input()
z = raw_input()
output(float(x), float(y),float(z))
```

Chiedi all' utente due numeri, se vuole invertire l' ordine di inserimento e stampa nell' ordine richiesto.

```
x = raw_input("scrivi due numeri")
y = raw_input()
z = raw_input("vuoi invertire l'ordine di inserimento? (se no inserisci 0, se si un altro intero)")
if int(z):
    print y, x
else:
    print x, y
```

Scrivere una procedura che chiede all' utente tre numeri e stampa il numero di valori distinti.

```
x = raw_input("scrivi tre numeri")
y = raw_input()
z = raw_input()
print "il numero di valori distinti è ",
if (x == y) and (y==z):
    print 1
elif (x == y) or (y==z) or (x==z):
    print 2
else:
    print 3
```

Scrivere una procedura che stampa il prezzo del biglietto di uno spettacolo in un club, secondo la regola: dieci euro se l' utente ha meno di sessanta anni e non è membro del club dove si tiene lo spettacolo, cinque se è un membro con meno di sessanta anni o, non essendo membro, ne ha almeno sessanta, tre, altrimenti.

```
età = raw_input("quanti anni hai? ")
membro = raw_input("sei un membro del club? (se no inserisci 0, se si un altro intero)")
print "il costo del biglietto è di euro ",
if (int(età) < 60) and not(int(membro)):
    print 10
elif (int(età) < 60) or not(int(membro)):
    print 5
else:
    print 3
```

Scrivere una procedura che svolge il seguente compito:

- chiede a un piccolo imprenditore il nome dei suoi tre dipendenti e il numero di ore che hanno lavorato durante il giorno;
- assumendo che la paga è di dieci euro l'ora, stampa il nome e la paga giornaliera di ognuno in ordine decrescente di retribuzione.

```
def confronto(x, orex, y, orey):  
    if orex > orey:  
        print x, 10*orex, "euro"  
        print y, 10*orey, "euro"  
    else:  
        print y, 10*orey, "euro"  
        print x, 10*orex, "euro"  
  
dip1 = raw_input("nome primo dipendente: ")  
ore1 = raw_input("numero ore: ")  
ore1 = int(ore1)  
dip2 = raw_input("nome secondo dipendente: ")  
ore2 = raw_input("numero ore: ")  
ore2 = int(ore2)  
dip3 = raw_input("nome terzo dipendente: ")  
ore3 = raw_input("numero ore: ")  
ore3 = int(ore3)  
  
if (ore1 > ore2) and (ore1 > ore3):  
    print dip1, 10*ore1, "euro"  
    confronto(dip2, ore2, dip3, ore3)  
  
elif (ore2 > ore3):  
    print dip2, 10*ore2, "euro"  
    confronto(dip1, ore1, dip3, ore3)  
  
else:  
    print dip3, 10*ore3, "euro"  
    confronto(dip1, ore1, dip2, ore2)
```

Scrivere una procedura che chiede tre numeri e stampa la somma usando una funzione produttrice.

```
def output(x,y,z):
    return x+y+z

x = raw_input("scrivi tre numeri")
y = raw_input()
z = raw_input()
print "la somma è: ", output(float(x), float(y), float(z))
```

Scrivere una procedura che chiede all'utente tre numeri interi diversi in ordine crescente.

```
def controllo_input(x):
    i = 0
    while i < len(x):
        if x[i] in "0123456789":
            i = i+1
        else:
            return 1
    return 0

def nuova_richiesta(x):
    while controllo_input(x):
        x=raw_input("devi digitare un intero: ")
    x = int(x)

x = raw_input("digita il primo numero intero: ")
nuova_richiesta(x)
y=raw_input("digita il secondo numero intero: ")
nuova_richiesta(y)
while y <= x:
    y=raw_input("il numero deve essere strettamente maggiore del precedente: ")
    nuova_richiesta(y)
z=raw_input("digita il terzo numero intero: ")
nuova_richiesta(z)
while z <= y :
    z=raw_input("il numero deve essere strettamente maggiore dei precedenti: ")
    nuova_richiesta(z)
```

Scrivere una procedura che memorizza in una lista una sequenza di interi inseriti dall'utente.

```
def controllo_input(x):
    if x == "q":
        return 0

    i = 0
    while i < len(x):
        if x[i] in "0123456789":
            i = i+1
        else:
            return 1

    return 0

def nuova_richiesta(x):
    while controllo_input(x):
        x=raw_input("devi digitare un intero o q per smettere: ")
    if x == "q":
        return
    else:
        x = int(x)

lista = []
x = raw_input("digita un numero intero o q per smettere: ")
nuova_richiesta(x)
while x != "q":
    lista = lista + [x]
    x = raw_input("digita un numero intero o q per smettere: ")
    nuova_richiesta(x)
```

Scrivere una procedura che copia in una lista i numeri dal file NUMERI.pkl.

```
import pickle
f = open("NUMERI.pkl", "r")
lista = []
x = pickle.load(f)
while x != "" :
    lista = lista + [x]
    x = pickle.load(f)
f.close()
```

Scrivere una procedura che legge i numeri dal file non vuoto NUMERI.pkl e stampa la loro somma.

```
import pickle
f = open("NUMERI.pkl", "r")
x = pickle.load(f)
somma = 0
while x != "":
    somma = somma + x
    x = pickle.load(f)
print somma
f.close()
```

Scrivere una procedura che legge i numeri dal file NUMERI.pkl e stampa il loro prodotto.

```
import pickle
f = open("NUMERI.pkl", "r")
x = pickle.load(f)
if x == "":
    prodotto = ""
else:
    prodotto = 1
while x != "":
    prodotto = prodotto * x
    x = pickle.load(f)
print prodotto
f.close()
```

Scrivere una procedura che legge i numeri dal file NUMERI.pkl e stampa il valore massimo.

```
import pickle
f = open("NUMERI.pkl", "r")
x = pickle.load(f)
massimo = x
while x != "":
    if massimo < x:
        massimo = x
    x = pickle.load(f)
print massimo
f.close()
```

Scrivere una procedura che legge i numeri dal file NUMERI.pkl e stampa la media dei valori.

```
import pickle
f = open("NUMERI.pkl", "r")
somma = 0
contatore=0
x = pickle.load(f)
while x != "" :
    contatore = contatore + 1
    somma = somma + x
    x = pickle.load(f)
if contatore > 0:
    print somma/contatore
f.close()
```

Il file DATI.pkl contiene l' informazione su un gruppo di ex-studenti. Per ognuno, è memorizzato il nome e l' anno di laurea. Scrivere una procedura che stampa il nome degli ex-studenti laureati nello scorso millennio.

```
import pickle
f = open("DATI.pkl", "r")
nome= pickle.load(f)
while nome != "" :
    anno_laurea = pickle.load(f)
    if anno_laurea < 2000 :
        print nome
    nome = pickle.load(f)
f.close()
```

Il file DATI.pkl contiene l'informazione su un gruppo di lavoratori. Per ognuno di essi, è memorizzato il nome e se è a tempo pieno (1 sì, 0 no). Scrivere una procedura che stampa il nome dei lavoratori a tempo pieno.

```
import pickle

f = open("DATI.pkl", "r")

nome= pickle.load(f)

while nome != "" :

    tempo_pieno = pickle.load(f)

    if tempo_pieno :

        print nome

    nome = pickle.load(f)

f.close()
```

Il file DATI.pkl contiene l'informazione su un gruppo di lavoratori. Per ognuno di essi, è memorizzato il nome e l'anno di assunzione. Scrivere una procedura che stampa il nome dei lavoratori che sono almeno al decimo anno di servizio, dopo aver chiesto all'utente il valore dell'anno corrente.

```
import pickle

f = open("DATI.pkl", "r")

nome= pickle.load(f)

anno_corrente = raw_input("inserisci anno corrente")

anno_corrente = int(anno_corrente)

while nome != "" :

    anno_assunzione = pickle.load(f)

    if anno_corrente - anno_assunzione >= 10 :

        print nome

    nome = pickle.load(f)

f.close()
```

Scrivere una procedura che legge il nome e anno di nascita di un personaggio dal file DATI.pkl e chiede all'utente di indovinare l'anno. Dopo al massimo tre tentativi gli viene chiesto se vuole giocare ancora.


```

import pickle

f = open("DATI.pkl", "r")

nome = pickle.load(f)

gioco = "sì"

while nome != "" and gioco == "sì":

    risposta = raw_input("quale è l'anno di nascita di " + nome + " ?")

    anno = pickle.load(f)

    i = 1

    while anno != risposta and i < 3 :

        risposta = raw_input("sbagliato, ritenta ")

        i = i + 1

    if risposta == anno :

        gioco=raw_input ("risposta esatta; se vuoi provare con il prossimo, inserisci sì")

    else:

        gioco=raw_input( "risposta sbagliata; se vuoi provare con il prossimo, inserisci sì")

    if gioco == "sì":

        nome = pickle.load(f)

if gioco == "sì":

    print "i nomi sono finiti"

f.close()

```

Chiedi all'utente di inserire l'anno di morte per ogni nome del file DATI.pkl in un nuovo file PERSONAGGI.pkl.

```

import pickle

f = open("DATI.pkl", "r")

g = open("PERSONAGGI.pkl", "w")

nome = pickle.load(f)

while nome != "":

    morte = raw_input("quale è l'anno di morte di " + nome + " ?")

    pickle.dump(nome, g)

    pickle.dump(pickle.load(f) , g)

    pickle.dump(morte, g)

    nome = pickle.load(f)

f.close()

g.close()

```

Il file DATI.pkl contiene l'informazione su un gruppo di lavoratori. Per ognuno, è memorizzato il nome e l'anno di assunzione. Scrivere una procedura che stampa l'anno di assunzione più remoto e il nome dei lavoratori assunti in tale anno.

```
import pickle

class Lavoratore:
    pass

f = open("DATI.pkl", "r")
lavoratore = Lavoratore()
lavoratore.nome = pickle.load(f)
lavoratore.assunzione = pickle.load(f)
minimo = lavoratore.assunzione
lavoratori = []
while lavoratore.nome != "" :
    if lavoratore.assunzione < minimo :
        minimo = lavoratore.assunzione
        lavoratori = lavoratori + [lavoratore]
        lavoratore.nome = pickle.load(f)
        lavoratore.assunzione = pickle.load(f)
print "anno assunzione più remota", minimo
for lavoratore in lavoratori:
    if lavoratore.assunzione == minimo:
        print lavoratore.nome
f.close()
```