

Esercizi - 002

ESERCIZIO 1

Data una costante chiamata BUDGET di 1000 euro, scrivere un algoritmo che prenda in **readline()** una cifra espressa in euro.

Gestire:

- Che la cifra non sia superiore a 1000 euro;
- In caso sia inferiore o uguale, dovrete dividere la cifra inserita in 3 RATE di pagamento e stampare in console un messaggio: "Paga in 3 comode rate da: \$valore_calcolato Euro";

TIPS: Dopo la divisione vi uscirà un numero float con tutta la parte decimale, per gestirla e limitare a 2 le cifre dopo la virgola, potete utilizzare la built-in function **round()** in questo modo: **round(\$valore_calcolato, 2);**

ESERCIZIO 2

Scrivere un algoritmo che prenda in input mediante il **readline()** un anno di nascita (numero intero composto da 4 cifre, solo anno senza specificare giorno o mese) e calcolare l'età effettiva con una sottrazione (ad es. : 2022 - 1992 = 30) .

Gestire:

- Caso in cui l'età calcolata sia un numero negativo, visualizzare un messaggio di errore "Il numero da te inserito non può essere elaborato";
 - Se l'età ha un valore **compreso** tra 0 e 3, voglio visualizzare il messaggio: "Troppo piccolo per scrivere a macchina. ";
 - Se l'età ha un valore **di 4 o superiore** , voglio visualizzare il messaggio: "Il nostro sistema ha calcolato che hai: Hai \$variabile_eta_calcolata anni";
-

ESERCIZIO 3

Utilizzando il costrutto Switch, scrivere un algoritmo per una pizzeria che prenda in input mediante il **readline()** una qualsiasi ordinazione (una stringa).

Stringa da Verificare - Costo

- Pizza - 6 Euro;
- Birra - 3 Euro;
- Panino - 8 Euro
- Insalata - Non Disponibile;
- Dessert - 3 Euro;

Scrivere tutte le casistiche qui sopra enunciate in cui, dato in input il nome del prodotto da ordinare, venga stampato in output una frase del tipo:

"Hai ordinato Carne con un costo di 10 Euro";

Gestire anche la casistica in cui venga inserita una pietanza non presente nel menu.

ESERCIZIO 4

Presi tutti i numeri da 1 a 10, scrivere un programma che calcoli la somma di tutti i numeri pari tralasciando quelli dispari;

TIPS: Per verificare se un numero è pari o dispari, dovete controllare se il resto della divisione per 2 sia 0 (FALSE) o 1 (TRUE).

```
2 Diviso 2 = 1 (con resto 0) // Numero Pari
```

```
3 Diviso 2 = 1 (con resto 1) // Numero Disapri
```

```
13 Diviso 2 = 6 (con resto 1) // Numero Disapri
```

In php potete utilizzare l'operatore **modulo %** per ottenere il resto della divisione, in questo modo:

```
(2 % 2) == 0 // True perche da resto 0 = Numero Pari
```

```
(3 % 2) == 0 // False perche da resto 1 = Numero Disapri
```

```
(13 % 2) == 0 // False perche da resto 1 = Numero Disapri
```

ESERCIZIO 5

Utilizzando la logica appena appresa con l'operatore Modulo, scrivere un programma che stampi in console tutti i numeri da 1 a 30.

- Se il numero e' multiplo di 3 deve stampare "PHP";
- Se multiplo di 5 deve stampare "JAVASCRIPT";
- Se multiplo di 3 e 5 (15) deve stampare "Aulab";

```
//Risultato
```

```
1,2,PHP,4,Javascript,PHP,7,8,PHP,Javascript,11,PHP,13,14,Aulab,16,17,PHP,19,  
Javascript,PHP,22,23,PHP,Javascript,26,PHP,28,29,Aulab
```

ESERCIZIO 6

Dato il seguente Array:

```
$arrays = [  
  [  
    'name' ⇒ 'Marco',  
    'gender' ⇒ 'M',  
  ],  
  [  
    'name' ⇒ 'Vanessa',  
    'gender' ⇒ 'F',  
  ],  
  [  
    'name' ⇒ 'Jack',  
    'gender' ⇒ 'M',  
  ],  
]
```

```
[  
  'name' ⇒ 'Mari*'  
  'gender' ⇒ '',  
],  
];
```

realizzare un ciclo **foreach** che:

- Se il genere e' M stampare "Buongiorno Sig. \$nome"
- Se e' F stampare "Buongiorno Sig.ra \$nome" .
- Se non è definito Stampare "\$nome ha preferito non specificare."

ESERCIZIO 7

Utilizzando il costrutto While (Pre-Condizionale), realizzare un sistema di **"Aggiungi al carrello"** in modo tale da eseguire delle operazioni di somma fino al raggiungimento del budget massimo inserito dall'utente appena eseguirete il programma:

```
//Ad Esempio  
  
$budget = 10;// valore preso in input da readline  
  
//Aggiungo articolo 1  
  
$costo_articolo_1 = 5;// valore preso in input da readline  
  
-----  
  
$budget = $budget - costo_articolo_1;  
  
//Budget Rimanente 5 Euro;  
  
//....
```

Nel costrutto while, una volta definita la condizione, dovrete di volta in volta chiedere all'utente il costo dell'articolo e visualizzare a schermo il budget

Rimanente.

Nel momento in cui il budget sarà 0 il programma terminerà l'esecuzione.

ATTENZIONE: Gestire l'eventualità in cui venga scritto un prezzo maggiore del budget a disposizione;

Il Budget rimanente non potrà mai assumere un valore negativo;

ESERCIZIO 8

Ripetere l'esercizio 7 con il costrutto Do While (Post Condizionale) ma attenzione:

- Gestire la condizione in cui il budget inserito sia 0, non deve essere possibile effettuare alcuna operazione;
- Fintantoché il budget inserito sia 0 o negativo, devo far visualizzare all'utente un messaggio di inserimento