

## Risposte Esame di Programmazione - 2021-12-22

### Esercizio 1

Realizzare una funzione che prenda in input un numero intero e restituisca il numero di cifre che lo compongono, ignorando il segno.

```
int contaCifre(int n) {  
    n = abs(n);  
    if (n < 10) {  
        return 1;  
    } else {  
        return 1 + contaCifre(n / 10);  
    }  
}
```

### Esercizio 2

Realizzare una funzione ricorsiva che operi su array statici di interi e restituisca il numero di elementi di A uguali a val.

```
int CountOccurrences(int A[], int size, int val) {  
    if (size == 0) {  
        return 0;  
    } else {  
        if (A[size - 1] == val) {  
            return 1 + CountOccurrences(A, size - 1, val);  
        } else {
```

## Risposte Esame di Programmazione - 2021-12-22

```
    return CountOccurrences(A, size - 1, val);  
}  
  
}  
  
}
```

### Esercizio 3A

Realizzare le strutture che descrivano:

- turista (codice fiscale, cognome, nome, nazionalita)
- insieme\_di\_turisti, realizzato tramite liste semplici.

```
struct Turista {  
    std::string codice_fiscale;  
    std::string cognome;  
    std::string nome;  
    std::string nazionalita;  
};
```

```
struct Nodo {  
    Turista t;  
    Nodo* next;  
};
```

```
struct Insieme_di_turisti {  
    Nodo* head;  
};
```

## Risposte Esame di Programmazione - 2021-12-22

### Esercizio 3B

Realizzare la funzione booleana `StessoTurista`, che ritorna `true` se due variabili di tipo `turista` sono uguali.

```
bool StessoTurista(Turista t1, Turista t2) {  
    return (t1.codice_fiscale == t2.codice_fiscale) && (t1.cognome == t2.cognome) && (t1.nome ==  
t2.nome) && (t1.nazionalita == t2.nazionalita);  
}
```

### Esercizio 3C

Produrre i prototipi (o interfacce) delle funzioni stando molto attenti ai parametri formali.

```
void InserisciTurista(Insieme_di_turisti &insieme, Turista t);  
void CancellaTurista(Insieme_di_turisti &insieme, std::string codice_fiscale);  
bool InsiemeVuoto(Insieme_di_turisti &insieme);  
void StampaInsiemeTuristi(Insieme_di_turisti &insieme);  
Insieme_di_turisti UnioneTraInsiemi(Insieme_di_turisti &insieme1, Insieme_di_turisti &insieme2);
```

### Esercizio 3D

Implementare la funzione `UnioneTraInsiemi`.

```
Insieme_di_turisti UnioneTraInsiemi(Insieme_di_turisti &insieme1, Insieme_di_turisti &insieme2) {  
    Insieme_di_turisti nuovoinsieme = insieme1;  
    Nodo* nodo = insieme2.head;
```

## Risposte Esame di Programmazione - 2021-12-22

```
while (nodo != NULL) {  
    if (!EsisteInInsieme(nuovoInsieme, nodo->t)) {  
        InserisciTurista(nuovoInsieme, nodo->t);  
    }  
    nodo = nodo->next;  
}  
return nuovoInsieme;  
}
```