

Quiz primo compitino

17/04/2024

Ho inserito in questo file LaTeX prima gli esercizi dei quiz in modo che vi possiate esercitare, le soluzioni sono inserite in fondo al pdf

1 Esercizi

1.1 Esercizio 1

(da completare se il numero di crediti del proprio esame è almeno 9)

Data la seguente funzione,

```
#include <stdio.h>
```

```
int f(int n) {
```

```
    if(n==1)
```

```
        return 1;
```

```
    else
```

```
        return 2*f(n-1);
```

```
}
```

scrivere la sua POSTcondizione;

scrivere la sua PREcondizione;

1.2 Esercizio 2

(da completare se i crediti del proprio esame sono almeno 6)

Specificare se il codice seguente compila e, nel caso, indicare cosa stampa ciascuna printf, motivando brevemente le risposte.

```
#include <stdio.h>
```

```
int x=1;
```

```
int main(void) {
```

```
    int m[5][2]={1,2,3,4,5,6,7,8,9,10};
```

```
    int *p;
```

```
    int *n = (int *) m;
```

```
    int **q = &p;
```

```
    {
```

```
        int x=2;
```

```
        p=&x;
```

```
        *p=*p/4*2;
```

```
        printf(" %d", x);           // 1)
```

```
    }
```

```
    printf(" %d", n[*p+1]);         // 2)
```

```
    printf(" %d\n", (*(m+2))[1]);   // 3)
```

```
}
```

1.3 Esercizio 3

(da completare se i crediti del proprio esame sono almeno 9)

Per ciascuno dei seguenti tipi, scriverne la dichiarazione (utilizzando in un unico comando/dichiarazione senza usare il carattere =):

1. x è un array di 5 elementi, ciascuno dei quali è un puntatore a carattere
2. p è un puntatore ad un array di 7 elementi, ciascuno dei quali è un puntatore ad numero reale in doppia precisione;

Esempio: per "y è un puntatore ad intero" scriveremmo: `int *y;`

1.4 Esercizio 4

(completare se i crediti del proprio esame sono almeno 6)

Data la seguente funzione

```
// PRE: soglia  $\geq 1$   
// POST: f(soglia) restituisce il minimo intero  $x \geq 1$  tale che  $x! > soglia$   
// (si ricordi che il fattoriale  $n!$  di un numero intero  $n \geq 1$  è definito come  $n*(n-1)*(n-1)*...*1$ , mentre  $0! = 1$ .  
// In alternativa possiamo definire  $n!$  ricorsivamente tramite:  $0! = 1$ ,  $(n+1)! = (n+1)*n!$  )
```

```
inf f(int soglia) {  
    int x = 1;  
    int fact = 1;  
    while (fact <= soglia) {  
        x += 1;  
        fact = x*fact;  
    }  
    return x;  
}
```

determinare quale delle seguenti asserzioni è una corretta invariante del ciclo while nel corpo di f.

INV1: $fact = x! \wedge x \geq 1 \wedge (x-1)! \leq soglia$

INV2: $fact = x! \wedge x \geq 1 \wedge x! \leq soglia$

INV3: $fact = (x+1)! \wedge x \geq 0 \wedge fact \leq soglia$

INV4: $fact = (x-1)! \wedge x \geq 0 \wedge fact \leq soglia$

2 Soluzioni

2.1 Soluzione esercizio 1

PRE: $n \geq 1$

POST: restituisce la potenza di 2 elevato alla $n-1$

2.2 Soluzione esercizio 2

//1: stampa 0 in quanto il puntatore p punta ad $x=2$, modificando il suo valore nella riga precedente per via dell'aritmetica del C, la quale esegue prima $2/4=0$ (troncando il valore ottenuto) moltiplicandolo per 2 ottenendo 0 nuovamente

//2: dangling reference, l'indice dell'array è indicato dal puntatore p il quale però punta ad una variabile deallocata, quindi indefinita

//3: stampa 6: infatti $*(m+2)$ punta al primo elemento della seconda riga ma con $*(m+2)[1]$ ci si riferisce al secondo elemento della seconda riga, ovvero 6

2.3 Soluzione esercizio 3

```
char *x[5]
```

```
double* (*p)[7]
```

2.4 Soluzione esercizio 4

INV1