

Domanda 1

Risposta errata

Punteggio
ottenuto 0,00 su
1,00Contrassegna
domanda

Si assuma un array `int Y[10][10]` e si consideri il seguente frammento di programma:

```
bool ok=true;
for(int i=0; i <10 && ok; i++)
{
    if(Y[i][1]!=Y[1][9-i])
        ok=false;
}
//POST
quale POST all'uscita dei cicli è appropriato?
```

Scegli un'alternativa:

- ☐ ok è vero sse la diagonale secondaria è uguale a quella principale.
- ☐ ok è vero sse la diagonale secondaria è l'inverso di quella principale.
- ☒ ok è vero sse la diagonale principale è un palindromo ❌

La risposta corretta è: ok è vero sse la diagonale secondaria è uguale a quella principale.

Domanda 2

Risposta
correttaPunteggio
ottenuto 1,00 su
1,00Contrassegna
domanda

Dato `int X[4][5][6][8]`, scegliere la risposta che specifichi tipo e valore dell'espressione:

`((*X) - 2)[3] - 1`.

Scegli un'alternativa:

- ☒ il tipo è `int(*)[8]` e il valore `X + 6*8*4 - 8*4` ✔️
- ☐ il tipo è `int(*)[8]` e il valore `X - 2*5*6*8*4 + 3*6*8*4 - 8*4`
- ☐ il tipo è `int*` e il valore `X + 6*8*4 - 4`

La risposta corretta è: il tipo è `int(*)[8]` e il valore `X + 6*8*4 - 8*4`

Domanda 3

Risposta errata

Punteggio
ottenuto 0,00 su
1,00



Contrassegna
domanda

Supponete di creare un albero BST partendo da un albero vuoto ed aggiungendo nodi con campo info uguali ai seguenti valori e inseriti nell'albero seguendo l'ordine specificato:

10 5 6 20 15 18 23 2 7 11

qual è l'albero BST ottenuto?

Scegli un'alternativa:

- ☐ 10(6(5(2(,),7(,)),15(11(,),20(18(,),23(,))))
- ☐ 10(5(2(,),6(,7(,))),20(15(11(,),18(,)),23(,)))
- ☒ 10(6(5(2(,),7(,)),20(18(15(11(,),23(,)))) ✖

La risposta corretta è: 10(5(2(,),6(,7(,))),20(15(11(,),18(,)),23(,)))

Domanda 4

Risposta
corretta

Punteggio
ottenuto 1,00 su
1,00



Contrassegna
domanda

Scegliere la risposta appropriata per la seguente funzione:

```
int *f (int *&x, int*y) {int** z=&y; *z=x; return *z;}
```

Scegli un'alternativa:

- ☒ La funzione restituisce un dangling reference ✔
- ☐ La funzione ha un errore di tipo
- ☐ La funzione è corretta

La risposta corretta è: La funzione restituisce un dangling reference

Domanda 5

Risposta
correttaPuntaggio
ottenuto 1,00 su
1,00Contrassegna
domanda

Supponiamo di avere un array `int y[5][10]` che è riempito per strati con `nle` elementi, $0 \leq nle \leq 250$. Scegliere la risposta con la formula il cui valore è il numero di elementi definiti della H-fetta `m`, $0 \leq m \leq 4$.

Scegli un'alternativa:

- ☐ $(nle/50)*10 + (m < (nle\%50)? 10:0) + (m == (nle\%10)/5 ? (nle/10):0)$
- ☐ $(nle/50)*10 + (m < (nle\%50)/10? 10:0) + (m == (nle\%50)? (nle\%10):0)$
- ☒ $(nle/50)*10 + (m < (nle\%50)/10? 10:0) + (m == (nle\%50)/10? (nle\%10):0)$ ✓

La risposta corretta è: $(nle/50)*10 + (m < (nle\%50)/10? 10:0) + (m == (nle\%50)/10? (nle\%10):0)$

Domanda 6

Parzialmente
correttaPuntaggio
ottenuto 0,20 su
1,00Contrassegna
domanda

Considerate la seguente funzione e scegliete le risposte appropriate. Nelle risposte `vLista(L)` è la lista `L` originale mentre `Lista(L)` è la lista `L` alla fine di `f`. Le posizioni dei nodi di una lista sono numerate `0,1,2,...` eccetera. Si assuma che `n` e `k` siano > 0 .

```
nodo* f(nodo* L, int k, int n)
{
    if (!L) return 0;
    if (n%k == 0)
        return f(L->next, k, n-1);
    else
    {
        nodo* s = L;
        L = L->next;
        s->next = f(L, k, n-1);
        return s;
    }
}
```

Scegli una o più alternative:

- ☐ se `vLista(L)` contiene più di `n%k` nodi, allora la funzione restituisce col return i nodi di `vLista(L)` dalla posizione `n%k+1` in poi
- ☐ se `vLista(L)` contiene `n%k+1+b` `k` nodi, con `b >= 0`, allora, la funzione restituisce col return `b+1` nodi
- ☐ se `vLista(L)` ha `n%k + b` `k` nodi, con `b >= 0`, allora alla fine della funzione `Lista(L)` ha `b` nodi meno di `vLista(L)`
- ☒ se `vLista(L)` ha non più di `n%k` nodi, allora la funzione restituisce 0 col return ✓
- ☐ se `n%k > 0` e `vLista(L)` ha `n%k` nodi, allora la funzione restituisce col return 1 nodo
- ☐ se `vLista(L)` contiene `n%k+1+b` `k`, con `b >= 0`, nodi, allora, alla fine della funzione, `Lista(L)` ha `n%k + b` nodi

Le risposte corrette sono: se `vLista(L)` contiene `n%k+1+b` `k` nodi, con `b >= 0`, allora, la funzione restituisce col return `b+1` nodi, se `vLista(L)` ha `n%k + b` `k` nodi, con `b >= 0`, allora alla fine della funzione `Lista(L)` ha `b` nodi meno di `vLista(L)`, se `vLista(L)` ha non più di `n%k` nodi, allora la funzione restituisce 0 col return

Domanda **7**
Risposta errata
Punteggio
ottenuto 0.00 su
1.00
Contrassegna
domanda

Si consideri la seguente funzione su alberi binari e si scelga la risposta appropriata. Assumete che $n \geq 0$, `Albero(r)` è ben formato.

```
bool f(nodo*r, int n)
{
    if(!r) return false;
    if(n==0)
        if(!r->left && !r->right)
            return true;
        else
            return false;
    else
        return f(r->left,n-1) && f(r->right,n-1);
}
```

Scegli un'alternativa:

- ☒ la funzione restituisce true sse `albero(r)` ha tutte le foglie a profondità maggiore o uguale a n ❌
- ☐ la funzione restituisce true sse `albero(r)` ha tutte le foglie a profondità n
- ☐ la funzione restituisce true sse `albero(r)` è un albero completo di profondità n

La risposta corretta è: la funzione restituisce true sse `albero(r)` è un albero completo di profondità n

Fine revisione