

Proposta di soluzione

a cura di
Stefano Cherubin*

prova del 13/02/2015

[**Informatica A**] Seconda prova in itinere, corso per Ing. Gestionale a.a.
2014/15

*<nome.cognome>@polimi.it

Indice

1	Analisi di codice	3
1.1	Osservazioni	3
1.2	Soluzione	3
2	Produzione di codice (struct)	4
2.1	Soluzione	4
3	Produzione di codice (matrici)	5
3.1	Soluzione	5

1 Analisi di codice

Si dica cosa stampa il seguente codice e si spieghi cosa calcola la funzione `f`.

```
void f(int a, int b) {
    if (!(b > 1 && b <= 10))
        return;
    if (a/b <= 0)
        printf("\n");
    else
        f(a/b, b);
    printf("%d", a % b);
    return;
}

int main() {
    int i = 0;
    char c;
    int v1[5] = {7, 16, 34, 4, 2};
    int v2[5] = {2, 8, 6, 2, 2};
    for(i = 0; i < 5; i++) {
        f(v1[i], v2[i]);
    }
    scanf("%c", &c);
    return 0;
}
```

1.1 Osservazioni

La funzione `f` è ricorsiva. Essa prevede due casi base e un passo induttivo:

- caso base: $b \notin [2; 10]$, non esegue nulla
- caso base: $a/b \leq 0$, va a capo, poi stampa `a%b`
- passo induttivo: $a/b > 0$, esegue `f` su `a/b`, `b`

La funzione `main` dichiara e inizializza 2 vettori, ciascuno di 5 elementi, e chiama la funzione una volta per ogni elemento dei vettori.

1.2 Soluzione

La funzione `f` esegue in modo ricorsivo l'algoritmo di cambio di base per divisioni successive. L'algoritmo non gestisce basi maggiori di 10.

Il programma fornisce in output la codifica di 7 in base 2; 16 in base 8; 34 in base 6; 4 in base 2 e 2 in base 2.

```
111 20 54 100 10
```

2 Produzione di codice (struct)

Si consideri la seguente definizione di punto:

```
typedef struct {  
    int x;  
    int y;  
} punto;
```

Scrivere una funzione f che riceve in input:

- una matrice M di interi
 - di dimensione $N \times N$, con N costante predefinita da una **#define**
- un vettore di punti
 - di dimensione N

```
int f(int M[][N], punto punti[N]);
```

La funzione deve restituire la somma di tutti i valori della matrice le cui coordinate sono indicate nel vettore di punti e che risultano essere coordinate valide della matrice.

2.1 Soluzione

```
1 int f(int M[][N], punto punti[N]) {  
2     int i, somma;  
3     somma = 0;  
4     for (i = 0; i < N; i++) {  
5         if (punti[i].x >= 0 && punti[i].x < N &&  
6             punti[i].y >= 0 && punti[i].y < N) {  
7             somma += M[punti[i].x][punti[i].y];  
8         }  
9     }  
10    return somma;  
11 }
```

3 Produzione di codice (matrici)

Implementare la funzione **#define** che riceve in input due matrici:

- A
 - di dimensione NxN, con N costante predefinita da una **#define** di valore pari
- B
 - di dimensione N/2

```
int riduci (int A[][N], int B[][N/2])
```

La funzione calcola i valori di B partendo da quelli di A nel seguente modo: ciascun punto di B sarà calcolato come la media di quattro punti adiacenti di A, come mostrato in figura:

a_{00}	a_{01}	b_{00}	b_{01}
a_{10}	a_{11}	b_{10}	b_{11}
c_{00}	c_{01}	d_{00}	d_{01}
c_{10}	c_{11}	d_{10}	d_{11}

a	b
c	d

1	2	100	100
3	4	10	2
10	10	20	1
12	12	10	2

2	53
11	8

3.1 Soluzione

```
1 int riduci (int A[][N], int B[][N /2]) {
2     int i, j;
3     for (i = 0; i < N/2; i++) {
4         for (j = 0; j < N/2; j++) {
5             B[i][j] = A[i * 2][j * 2];
6             B[i][j] += A[i * 2 + 1][j * 2];
7             B[i][j] += A[i * 2][j * 2 + 1];
8             B[i][j] += A[i * 2 + 1][j * 2 + 1];
9             B[i][j] /= 4;
10        }
11    }
12    return 0;
13 }
```

Licenza e crediti

Crediti

Quest'opera contiene elementi tratti da materiale di Alessandro Campi redatto per il corso di Informatica A per Ingegneria Gestionale a.a. 2014/15.

Licenza beerware¹

Quest'opera è stata redatta da Stefano Cherubin. Mantenendo questa nota, puoi fare quello che vuoi con quest'opera. Se ci dovessimo incontrare e tu ritenessi che quest'opera lo valga, in cambio puoi offrirmi una birra.

¹<http://people.freebsd.org/~phk/>