

| | | | |
|--|---|---------|--|
|  | Politecnico di Milano Scuola di Ingegneria Industriale e dell'Informazione INFORMATICA B Seconda prova in itinere - 4 Febbraio 2016 | | COGNOME E NOME |
| | RIGA | COLONNA | MATRICOLA |
| | | | Spazio riservato ai docenti <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> |

- Il presente plico contiene 4 esercizi e **deve essere debitamente compilato con cognome e nome, numero di matricola.**
- Il tempo a disposizione è di 1 ore e 30 minuti.
- Non separate questi fogli. Scrivete la soluzione solo sui fogli distribuiti, utilizzando il retro delle pagine in caso di necessità. Cancellate le parti di brutta con un tratto di penna.
- Ogni parte non cancellata a penna sarà considerata parte integrante della soluzione.
- È possibile scrivere a matita (e non occorre ricalcare al momento della consegna!).
- **È vietato utilizzare telefoni o pc.** Chi tenti di farlo vedrà annullata la sua prova.
- È ammessa la consultazione di libri e appunti.
- **Qualsiasi tentativo di comunicare con altri studenti comporta l'espulsione dall'aula.**
- È possibile ritirarsi senza penalità.
- **Non è possibile lasciare l'aula conservando il tema della prova in corso.**

Esercizio 1 (5 punti)

Un array di interi, con un numero pari di elementi, si dice a *somma costante* se sommando a coppie gli elementi in posizione simmetrica rispetto al centro dell'array (cioè il primo con l'ultimo, il secondo con il penultimo, e così via) si ottiene sempre lo stesso risultato.

Scrivere un programma che:

- chieda all'utente di inserire da tastiera un intero n , pari, maggiore di zero e minore o uguale di 100 (si controlli che n rispetti queste condizioni e, in caso contrario, si re-iteri la richiesta di inserire n , fino all'inserimento di un valore ammissibile);
- chieda all'utente di inserire da tastiera un array di n interi
- verifichi se l'array inserito è a *somma costante* e stampi a video un messaggio per informare l'utente sull'esito della verifica

Soluzione

```
#include <stdio.h>

#define N 100

int main()
{
    int i, n, somma, cost=1, a[N];

    do
    {
        scanf("%d", &n);
    } while (n<=0 || n>100 || n%2 == 1);

    for (i=0; i<n; i++)
        scanf("%d", &a[i]);

    somma = a[0]+a[n-1];

    for (i=1; i<n/2; i++)
    {
        if (a[i]+a[n-i-1] != somma)
        {
            cost = 0;
            break;
        }
    }

    if (cost==1)
        printf("Array a somma costante\n");
    else
        printf("Array non a somma costante\n");

    return 0;
}
```

Esercizio 2 (6 punti)

Un operatore telefonico è caratterizzato da un nome, il paese in cui opera e le informazioni riguardanti i piani tariffari che offre (un operatore offre al massimo 20 piani differenti). Ogni piano tariffario, a sua volta, è caratterizzato da un nome, il costo mensile del piano, il numero di minuti mensili inclusi nel piano e il numero di GB di traffico dati mensili inclusi nel piano.

- dichiarare un tipo di dato in C in grado di rappresentare tutte le informazioni riguardanti un operatore telefonico (secondo quanto descritto in precedenza);
- si dichiarino due variabili, `db` e `sel`, in grado di contenere i dati di 100 operatori telefonici ciascuna;
- scrivere un frammento di codice in C che, ipotizzando che la variabile `db` sia stata precedentemente riempita con i dati di 100 operatori, copi nella variabile `sel` (senza lasciare spazi) i dati degli operatori che hanno almeno 3 piani tariffari con più di 2 GB di traffico dati inclusi;
- scrivere un frammento di codice C che stampi a video la percentuale degli operatori contenuti in `sel` che operano in Italia.

Note

Non occorre scrivere il frammento di codice necessario a caricare i dati nella variabile `db`. E' possibile svolgere i punti C e D insieme, tuttavia si suggerisce di mantenere separate le soluzioni dei due punti. Non è necessario scrivere un programma completo né includere le librerie utilizzate, ma è necessario includere nei frammenti di codice le dichiarazioni di tutte le variabili utilizzate.

Soluzione

a)

```
typedef struct {
    char nome[50];
    float canone;
    int minuti;
    int dati;
} piano;

typedef struct {
    char nome[50];
    char paese[50];
    int npiani;
    piano piani[20];
} operatore;
```

b)

```
operatore db[100], sel[100];
```

c)

```
int i, j, k, n;

k = 0;
for (i=0; i<100; i++){
    n=0;
    for (j=0; j<db[i].npiani; j++)
        if (db[i].piani[j].dati > 2)
            n++;
    if (n>=3) {
        sel[k] = db[i];
        k++;
    }
}
```

d)

```
n=0;
for (i=0; i<k; i++)
    if (strcmp(sel[i].paese, "Italia")==0)
        n++;

printf("La percentuale degli operatori italiani è: %f\n", (n*100.0)/k);
```


Esercizio 3 (3 punti)

- Si determini la codifica del valore 48.15 secondo lo Standard IEEE 754-1985 a precisione singola, riportando i calcoli effettuati.
- La codifica di 48.15, calcolata al punto A, è esatta? Giustificare la risposta.

Soluzione

a)

Parte intera: $48 \rightarrow 110000$

Parte frazionaria: 0.15

$$0.15 * 2 = 0.3$$

$$0.3 * 2 = 0.6$$

$$0.6 * 2 = 1.2$$

$$0.2 * 2 = 0.4$$

$$0.4 * 2 = 0.8$$

$$0.8 * 2 = 1.6$$

$$0.6 * 2 = 1.2 \dots \rightarrow 0.001001100110011001\dots \rightarrow 0.00(1001)$$

$$48.15 \rightarrow 1.1000000(1001) \times 2^5$$

Codifica finale

S:0

E:10000100

M: 10000001001100110011001

b)

La codifica di non è esatta dato che la mantissa è periodica.

Esercizio 4 (3 punti)

Su BestMemory.com sono disponibili due sistemi in offerta allo stesso prezzo che differiscono solo per il loro sistema di memoria.

Il **sistema A** è dotato di memoria centrale e memoria cache con le seguenti caratteristiche:

- Hit Rate = 75%
- Hit Time = 20ns
- Miss Penalty = 400ns

Il **sistema B** è dotato di memoria centrale senza memoria cache con un tempo medio di accesso di 100ns.

Rispondere alle seguenti domande:

- a) Quale dei due sistemi in offerta acquistereste? Giustificare la risposta.
- b) Cambierebbe la vostra risposta se l'Hit Rate del sistema A fosse pari all'80% ?

Nota: Con Miss Penalty si intende il tempo complessivo per accedere al dato in caso non si trovi nella cache (comprende perciò sia il tempo per accedere alla cache sia il tempo necessario per accedere al dato nella memoria centrale e per trasferirlo dalla memoria centrale alla cache)

Soluzione

- a) Il tempo medio di accesso del sistema A è:
 $T = 0.75 \cdot 20\text{ns} + 0.25 \cdot 400\text{ns} = 115\text{ns} \rightarrow$ conviene acquistare il sistema B
- b) Se $HR = 0.8$:
 $T = 0.8 \cdot 20\text{ns} + 0.2 \cdot 400\text{ns} = 96\text{ns} \rightarrow$ conviene acquistare il sistema A