	Politecnico di Milano Facoltà di Ingegneria Industriale INFORMATICA B Prova in itinere del 23 Novembre 2016		COGNOME E NOME					
	RIGA	COLONNA	MATRICOLA					
Spazio riservato ai docenti				<table border="1"> <tr> <td> </td> <td> </td> <td> </td> <td> </td> </tr> </table>				

- Il presente plico contiene **3 esercizi** e deve essere debitamente compilato con cognome e nome, numero di matricola e posizione durante lo scritto (comunicata dal docente).
- Il tempo a disposizione è di 1 ora 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** (o ripudiate) 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) assicurandosi comunque che **quanto scritto sia ben leggibile**.
- È **vietato** utilizzare **calcolatrici, telefoni o pc**. Chi tenti di farlo vedrà **annullata** la sua prova.
- È ammessa la consultazione di **libri e appunti**, purché con pacata discrezione e senza disturbare.
- **Non è tuttavia possibile utilizzare i temi d'esame svolti degli anni precedenti**.
- 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)

I numeri di Fibonacci sono una successione di numeri naturali in cui ciascun numero, rappresentato dal simbolo $F(n)$, è definito nel modo seguente:

se $n == 0$ oppure $n == 1$, $F(n) = 1$
altrimenti, $F(n) = F(n-1) + F(n-2)$

Data questa definizione, si sviluppi in linguaggio C un programma che dichiari un array di 20 elementi interi, e assegni agli elementi di questo array i primi valori della successione di Fibonacci. Per esempio, se a è la variabile di tipo array, $a[5]$ conterrà il valore di $F(5)$, pari a 8. Infine, si faccia in modo che il programma stampi a video tutti i valori nell'array che sono dispari. Ad esempio, si stampi:

$F(0) = 1$
 $F(1) = 1$
 $F(3) = 3$
 $F(4) = 5$
etc...

Soluzione

```
#include <stdio.h>
#define NVALORI 20
int main()
{
    int array[NVALORI];
    int i;
    array[0] = 1;
    array[1] = 1;
    printf("F(%d) = %d\n", 0, array[0]);
    printf("F(%d) = %d\n", 1, array[1]);
    for(i = 2; i < NVALORI; i++)
    {
        array[i] = array[i-1] + array[i-2];
        if(array[i] %2 != 0)
            printf("F(%d) = %d\n", i, array[i]);
    }
    return 0;
}
```

Esercizio 2 (8 punti)

Si consideri il seguente problema: il gestore di una casa di cura con 100 stanze vuole aumentare il livello di automazione dell'impianto di climatizzazione. Egli quindi dota ciascuna stanza di un numero variabile di sensori di temperatura (da zero fino ad un massimo di sei) e di un attuatore, il quale controlla l'accensione e lo spegnimento dell'impianto di climatizzazione. Si considerino le seguenti strutture in linguaggio C che modellano la situazione sopra descritta:

```
#define NSENSORI 6

typedef enum {spento, acceso} StatoAttuatore;

typedef struct {
    float datiSensori[NSENSORI];
    int nSensori;
    StatoAttuatore attuatore;
} Stanza;
```

dove l'array *datiSensori* contiene l'ultimo valore di temperatura (in gradi centigradi) misurato da ciascun sensore di temperatura della stanza, *nSensori* indica il numero di sensori effettivamente presenti in *datiSensori*, e *attuatore* rappresenta lo stato dell'attuatore nella stanza. Si risponda alle seguenti domande utilizzando il linguaggio di programmazione C:

Punto 1: Si definisca il tipo *CasaDiCura* atto a contenere esattamente 100 stanze, e si dichiari la variabile *smartCasa* che rappresenta una casa di cura;

Punto 2: Si assuma che la variabile *smartCasa* sia stata dichiarata e opportunamente riempita, per tutte le 100 stanze della casa di cura, con i valori relativi al numero di sensori presenti in ogni stanza e alle loro misurazioni di temperatura. Si scriva quindi un frammento di programma che, per ogni stanza della casa di cura, calcoli la temperatura minima e massima fornita dai sensori presenti e:

- a. se la temperatura minima è minore di 16 gradi centigradi, assegni all'attuatore della stanza il valore acceso;
- b. In caso contrario, se la temperatura massima è maggiore di 23, assegni all'attuatore della stanza il valore spento.

Punto 3: Si dichiari un'opportuna variabile *stanzeSpente* per contenere tutte le stanze aventi l'attuatore con stato spento, e si scriva un frammento di codice per popolare (senza lasciare buchi) la variabile *stanzeSpente* a partire dalla variabile *smartCasa*.

Punto 4: Il gestore delle casa di cura decide di installare in ogni stanza tre nuovi sensori che rilevano la presenza di oggetti in movimento (sensori di movimento). Ciascuno di questi sensori può assumere uno tra due valori possibili che indicano la presenza e l'assenza di movimento. Si estenda la struct *Stanza* in modo che includa anche questi nuovi sensori.

Nota: per ogni punto si dichiarino tutte le variabili necessarie a sviluppare il codice richiesto. Non è necessario scrivere un singolo programma che risponda alle richieste di tutti i punti, ma solamente delle porzioni di codice che soddisfino ognuna le singole richieste.

Soluzione

Punto 1)

```
#define NSTANZE 100
typedef Stanza CasaDiCura[NSTANZE];
CasaDiCura smartCasa;
```

Punto 2)

```
#define SOGLIAMAX 23
#define SOGLIAMIN 16
int main() {
    CasaDiCura smartCasa;
    int i, j;
    float tmin, tmax;

    /* ... istruzioni necessarie per inizializzare la struttura dati ... */
    for(i = 0; i < NSTANZE; i++) {
        tmin = 0;
        tmax = 0;
        for(j = 0; j < smartCasa[i].nSensori; j++)
        {
            if (smartCasa[i].datiSensori[j] < tmin)
                tmin = smartCasa[i].datiSensori[j];
            if (smartCasa[i].datiSensori[j] > tmax)
                tmax = smartCasa[i].datiSensori[j];
        }
        if (tmin <= SOGLIAMIN)
            smartCasa[i].attuatore = acceso;
        else if (tmax >= SOGLIAMAX)
            smartCasa[i].attuatore = spento;
    }
}
}
```

Punto 3)

```
Albergo stanzeSpente;
int nStanzeSpente = 0;
for(i = 0; i < NSTANZE; i++)
    if(smartCasa[i].attuatore == spento)
    {
        stanzeSpente[nStanzeSpente] = smartCasa[i];
        nStanzeSpente++;
    }
}
```

Punto 4)

```
#define N_SENSORI_MOV 3
typedef enum {falso, vero} Booleano;
typedef struct {
    float datiSensori[NSENSORI];
    int nSensori;
    Booleano sensoriMovimento[N_SENSORI_MOV];
    int nSensMov;
    StatoAttuatore attuatore;
} Stanza;
```

Esercizio 3 (4 punti)

Dati i due numeri interi -63 e 8 in codifica decimale,

A) si fornisca la rappresentazione in complemento a due di entrambi i numeri, stabilendo prima il minimo numero di bit in grado di rappresentarli entrambi.

B) si esegua la somma della codifica in complemento a due dei due numeri

C) si verifichi che il risultato della somma sia corretto verificando che il valore della sua codifica decimale sia uguale al valore della somma delle codifiche decimali dei due numeri interi dati

Soluzione

A)

Il numero di bit minimo per rappresentare entrambi deve essere il più piccolo numero naturale m t.c.: $-2^{m-1} \leq -63$ e $8 \leq 2^{m-1}-1$. In questo caso $m = 7$.

La rappresentazione del numero decimale 8_{dec} in codifica binaria naturale è 1000_{bin} e quindi, aggiungendovi il bit di segno, in complemento a due a 7 bit è 0001000_{C2} .

Dovendo rappresentare il numero -63_{dec} in complemento a due, considero prima il numero positivo 63_{dec} la cui codifica binaria naturale è pari a 111111_{bin} . Pertanto, il complemento a due di -63_{dec} a 7 bit è $(000000_{bin}+1_{bin}=000001_{bin}) 1000001_{C2}$.

B)

-63_{dec} : 1000001_{C2}

8_{dec} : 0001000_{C2}


01001001_{C2}

La codifica del risultato deve essere fornita su un numero di bit uguale a quello degli addendi, quindi su 7 bit. Il bit in posizione 8 e' il bit di carry che deve essere tolto dalla rappresentazione e quindi il risultato è: 1001001_{C2}

C)

Il bit in settima posizione è 1 e quindi indica segno -. I rimanenti 6 bit (001001) rappresentano il modulo del numero che in codifica decimale e' uguale a $0 + 0 + 8 + 0 + 0 + 1 = 9_{dec}$ che è uguale a

$-64_{dec} + 9_{dec} = -55_{dec}$ verificando pertanto il risultato della somma delle codifiche in complemento a due dei due numeri.

	Politecnico di Milano Facoltà di Ingegneria Industriale INFORMATICA B Prova in itinere del 23 Novembre 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 **3 esercizi** e deve essere debitamente compilato con cognome e nome, numero di matricola e posizione durante lo scritto (comunicata dal docente).
- Il tempo a disposizione è di 1 ora 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** (o ripudiate) 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) assicurandosi comunque che **quanto scritto sia ben leggibile**.
- È **vietato** utilizzare **calcolatrici, telefoni o pc**. Chi tenti di farlo vedrà **annullata** la sua prova.
- È ammessa la consultazione di **libri e appunti**, purché con pacata discrezione e senza disturbare.
- **Non è tuttavia possibile utilizzare i temi d'esame svolti degli anni precedenti**.
- 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)

Il fattoriale di un numero naturale n , rappresentato dal simbolo $n!$, è definito nel modo seguente:

se $n == 0$ oppure $n == 1$, $n! = 1$
altrimenti, $n! = n * (n - 1)!$

Data questa definizione, si sviluppi in linguaggio C un programma che dichiari un array di 15 elementi interi e che assegni a ciascun elemento dell'array il fattoriale del valore corrispondente all'indice dell'elemento stesso. Per esempio, se a è la variabile di tipo array, $a[5]$ conterrà il valore $5!$ (fattoriale di 5, pari a 120). Infine, si faccia in modo che il programma stampi a video tutti i valori nell'array che sono multipli di 5. Ad esempio, si stampi:

$5! = 120$
 $6! = 720$
... etc.

Soluzione

```
#include <stdio.h>
#define NVALORI 15
int main()
{
    int array[NVALORI];
    int i;
    array[0] = 1;
    array[1] = 1;
    for(i = 2; i < NVALORI; i++)
    {
        array[i] = i * array[i-1];
        if(array[i] %5 == 0)
            printf("%d! = %d\n", i, array[i]);
    }
    return 0;
}
```

Esercizio 2 (8 punti)

Si consideri il seguente problema: il gestore di un albergo con 500 stanze vuole aumentare il livello di automazione dell'impianto di climatizzazione. Egli quindi dota ciascuna stanza di un numero variabile di sensori di temperatura (da zero fino ad un massimo di sei) e di un attuatore, il quale controlla l'accensione e lo spegnimento dell'impianto di climatizzazione. Si considerino le seguenti strutture in linguaggio C che modellano la situazione sopra descritta:

```
#define NSENSORI 6

typedef enum {spento, acceso} StatoAttuatore;

typedef struct {
    float datiSensori[NSENSORI];
    int nSensori;
    StatoAttuatore attuatore;
} Stanza;
```

dove l'array *datiSensori* contiene l'ultimo valore di temperatura (in gradi centigradi) misurato da ciascun sensore di temperatura della stanza, *nSensori* indica il numero di sensori effettivamente presenti in *datiSensori*, e *attuatore* rappresenta lo stato dell'attuatore nella stanza. Si risponda alle seguenti domande utilizzando il linguaggio di programmazione C:

Punto 1: Si definisca il tipo *Albergo* atto a contenere esattamente 500 stanze, e si dichiari la variabile *smartHotel* che rappresenta un albergo;

Punto 2: Si assuma che la variabile *smartHotel* sia stata dichiarata e opportunamente riempita, per tutte le 500 stanze dell'albergo, con i valori relativi al numero di sensori presenti in ogni stanza e alle loro misurazioni di temperatura. Si scriva quindi un frammento di programma che, per ogni stanza dell'albergo, calcoli la temperatura media fornita dai sensori presenti e:
se questa è compresa tra 18 e 21 gradi centigradi (estremi inclusi), assegni all'attuatore della stanza il valore spento;
nel caso contrario, assegni all'attuatore della stanza il valore acceso.

Punto 3: Si dichiari un'opportuna variabile *stanzeAccese* per contenere tutte le stanze aventi l'attuatore con stato acceso, e si scriva un frammento di codice per popolare (senza lasciare buchi) la variabile *stanzeAccese* a partire dalla variabile *smartHotel*.

Punto 4: Si definisca un nuovo tipo *Appartamento* composto da un numero variabile di stanze (fino a un massimo di 5) e da un campo alfanumerico che definisce il cognome del proprietario.

Nota: per ogni punto si dichiarino tutte le variabili necessarie a sviluppare il codice richiesto. Non è necessario scrivere un singolo programma che risponda alle richieste di tutti i punti, ma solamente delle porzioni di codice che soddisfino ognuna le singole richieste.

Soluzione

Punto 1

```
#define NSTANZE 500
typedef Stanza Albergo[NSTANZE];
Albergo smartHotel;
```

Punto 2

```
#define SOGLIAMAX 21
#define SOGLIAMIN 18
int main() {
    Albergo smartHotel;
    int i, j;
    float media;

    /* ... istruzioni necessarie per inizializzare la struttura dati ... */

    for(i = 0; i < NSTANZE; i++) {
        media = 0;
        for(j = 0; j < smartHotel[i].nSensori; j++)
            media = media + smartHotel[i].datiSensori[j];
        media = media / smartHotel[i].nSensori;
        if(media <= SOGLIAMAX && media >= SOGLIAMIN)
            smartHotel[i].attuatore = spento;
        else
            smartHotel[i].attuatore = acceso;
    }
}
```

Punto 3

```
Albergo stanzeAccese;
int nStanzeAccese = 0;

for(i = 0; i < NSTANZE; i++) {
    if(smartHotel[i].attuatore == acceso) {
        stanzeAccese[nStanzeAccese] = smartHotel[i];
        nStanzeAccese++;
    }
}
```

Punto 4

```
#define MAX_LOCALI 5
#define MAX_LEN 30;

typedef char Stringa[MAX_LEN];

typedef struct{
    Stanza locali[MAX_LOCALI];
    int nLocali;
    Stringa cognome;
} Appartamento;
```

Esercizio 3 (4 punti)

Dati i due numeri interi 64 e -7 in codifica decimale,

A) si fornisca la rappresentazione in complemento a due di entrambi i numeri, stabilendo prima il minimo numero di bit in grado di rappresentarli entrambi.

B) si esegua la somma della codifica in complemento a due dei due numeri

C) si verifichi che il risultato della somma sia corretto verificando che il valore della sua codifica decimale sia uguale al valore della somma delle codifiche decimali dei due numeri interi dati)

Soluzione

A)

Il numero di bit minimo per rappresentare entrambi deve essere il più piccolo numero naturale m t.c.: $-2^{m-1} \leq -7$ e $64 \leq 2^{m-1}-1$. In questo caso $m = 8$.

La rappresentazione del numero decimale 64_{dec} in codifica binaria naturale è 1000000_{bin} e quindi, aggiungendovi il bit di segno, in complemento a due a 8 bit è 01000000_{C2} .

Dovendo rappresentare il numero -7_{dec} in complemento a due, considero prima il numero positivo 7_{dec} la cui codifica binaria naturale è pari a 111_{bin} . Pertanto 7_{dec} in codifica binaria naturale con 7 bit è 0000111_{bin} e il complemento a due di -7_{dec} ad 8 bit è $(1111000_{bin}+1_{bin}=1111001_{bin})$ 11111001_{C2} .

B)

64_{dec} : 01000000_{C2}

-7_{dec} : 11111001_{C2}

$[0](1)00111001_{C2}$

La codifica del risultato deve essere fornita su un numero di bit uguale a quello degli addendi, quindi su 8 bit. Il bit in posizione 9 e' il bit di carry che deve essere tolto dalla rappresentazione e quindi il risultato è: 00111001_{C2}

C)

Il bit in ottava posizione è 0 e quindi indica segno +. I rimanenti 7 bit (0111001) rappresentano il modulo del numero che in codifica decimale e' uguale a $0 + 32 + 16 + 8 + 0 + 0 + 1 = 57_{dec}$ che è uguale a $64_{dec} - 7_{dec} = 57_{dec}$ verificando pertanto il risultato della somma delle codifiche in complemento a due dei due numeri.