

	Politecnico di Milano Facoltà di Ingegneria Industriale INFORMATICA B Prova in itinere del 26 Novembre 2014		COGNOME E NOME				
	RIGA	COLONNA	MATRICOLA				
			Spazio riservato ai docenti <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></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 20 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 consultare temi d'esame 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.

TEMA A Esercizio 1 (4 punti)

Si consideri la seguente espressione booleana:

$$(((\text{NOT } A) \text{ OR } B) \text{ AND } (A \text{ AND } (\text{NOT } B))) \text{ OR } C$$

1. Si compili la seguente **tabella della verità** (in cui 0 rappresenta il valore logico FALSO, 1 il valore VERO):

A	B	C	NOT A	(NOT A) OR B	NOT B	A AND (NOT B)	((NOT A) OR B) AND (A AND (NOT B))	((NOT A) OR B) AND (A AND (NOT B)) OR C
0	0	0						
0	0	1						
0	1	0						
0	1	1						
1	0	0						
1	0	1						
1	1	0						
1	1	1						

2. Si consideri ora la condizione, scritta in linguaggio C, in cui **x** e **y** siano due variabili int:

$$((!(x<0) \parallel y>0) \&\& (x<0 \&\& !(y>0))) \parallel x>3$$

ottenuta dall'espressione booleana sopra indicata sostituendo la variabile A con **x<0**, la variabile B con **y>0**, la variabile C con **x>3**

Si risponda alle seguenti domande:

- L'espressione è vera o falsa quando $x=5$ e $y=7$? (giustificare la risposta)
- Se $x<0$, per quali valori di y l'espressione è vera? (giustificare la risposta)

Soluzione

1. L'intera espressione equivale a C

A	B	C	NOT A	(NOT A) OR B	NOT B	A AND (NOT B)	(NOT A) OR B) AND (A AND (NOT B))	((NOT A) OR B) AND (A AND (NOT B)) OR C
0	0	0	1	1	1	0	0	0
0	0	1	1	1	1	0	0	1
0	1	0	1	1	0	0	0	0
0	1	1	1	1	0	0	0	1
1	0	0	0	0	1	1	0	0
1	0	1	0	0	1	1	0	1
1	1	0	0	1	0	0	0	0
1	1	1	0	1	0	0	0	1

2. A) Quando $x=5 \rightarrow C=1 \rightarrow$ l'espressione è vera

B) Se $x < 0 \rightarrow C = 0 \rightarrow$ l'espressione è falsa per qualsiasi valore di y

Esercizio 2 (7 punti)

Si considerino le seguenti strutture dati, che permettono di memorizzare le informazioni di 20 piloti che partecipano ad una gara automobilistica di 50 giri. In particolare, vengono memorizzati il nome del pilota, il nome della sua squadra, il numero del pilota in gara, il numero di giri completati ed il tempo realizzato in ciascun giro completato.

```
typedef struct
{
    int minuti;
    int secondi;
    int millesimi;
} tempo;
```

```
typedef struct
{
    char nome[50];
    char squadra[50];
    int numero;
    int giri_completati;
    tempo giri[50];
} pilota;
```

```
pilota piloti[20];
```

- A. Si modifichino opportunamente le strutture dati qui sopra per memorizzare, per ogni pilota, anche:
 - a. il miglior tempo sul giro (cioè il tempo minore) effettuato da ciascun pilota nel corso della gara;
 - b. se il pilota ha effettivamente completato i 50 giri previsti dalla gara.
- B. Scrivere un frammento di programma C che, per ciascun pilota, calcoli le informazioni aggiuntive indicate nel punto precedente (miglior tempo sul giro e l'informazione sul completamento della gara).
- C. Scrivere un frammento di programma C che stampi a video nome, squadra e numero del pilota che (nel corso della gara) ha compiuto i 50 giri previsti e realizzato in assoluto il miglior tempo sul giro. Se nessun pilota ha completato tutti i giri, il frammento di programma dovrà stampare a video il messaggio "Gara ancora in corso, si prega di attendere".

Note: Si assuma che la variabile `piloti` sia già stata inizializzata (non è perciò richiesto di inizializzarla o acquisirla da tastiera). Per semplificare le operazioni richieste al punto B e C, si suggerisce di rappresentare il tempo migliore in secondi. Ad esempio, un tempo di 2 minuti, 4 secondi e 123 millesimi, può essere rappresentato con il suo valore in secondi: 124.123

Soluzione

1)

```
typedef struct
{
    char nome[50];
    char squadra[50];
    int numero;
    int giri_completati;
    tempo giri[50];
    float migliore;
    int ha_finito;
} pilota;
```

2)

```
int i, j;
float t, best;

for (i=0; i<20; i++)
{
    piloti[i].ha_finito = piloti[i].giri_completati == 50;
    best = piloti[i].giri[0].minuti*60+piloti[i].giri[0].secondi +
           piloti[i].giri[0].millesimi/1000.0;
    for (j=1; j<piloti[i].giri_completati; j++)
    {
        t = piloti[i].giri[j].minuti*60+piloti[i].giri[j].secondi +
           piloti[i].giri[j].millesimi/1000.0;
        if (t<best)
            best = t;
    }
}
```

```
        piloti[i].migliore = best;
    }
```

3)

```
int ii=0;

for (i=0; i<20; i++)
{
    if (piloti[i].migliore < piloti[ii].migliore && piloti[i].ha_finito ==
1)
        ii = i;
}
if (piloti[ii].ha_finito == 1)
printf("Il giro più veloce è stato realizzato da %s, numero %d, squadra
    %s\n",piloti[ii].nome,piloti[ii].numero,piloti[ii].squadra);
else printf("Gara ancora in corso, si prega di attendere\n");
```

Esercizio 3 (6 punti)

Si scriva un programma C per analizzare una matrice di interi M . La matrice deve essere definita con dimensioni 50×50 . Il programma dovrà:

1. Richiedere all'utente di inserire il numero di righe r e il numero di colonne c che desidera inserire, controllando che non superino le dimensioni massime consentite.
2. Richiedere l'inserimento di tutti gli elementi della matrice.
3. Scorrere la matrice per righe e copiare in un vettore v tutti i numeri pari fino ad un massimo di r elementi.
4. Stampare il contenuto di v a schermo e dire se la matrice contiene almeno r elementi pari.

N.B. Scorrere "la matrice per righe" vuol dire analizzare interamente la prima riga della matrice dal primo all'ultimo elemento, quindi scorrere allo stesso modo la seconda riga e così via fino all'ultima riga.

Esempio

12	3	1	2
8	4	7	22

 Se $M =$

il programma stamperà a schermo

$v = [12, 2]$

La matrice contiene almeno 2 elementi pari.

Soluzione

```
#include <stdio.h>
#define N 10

void main()
{
    int M[N][N];
    int vet[N*N];
    int r,c,i,j,flag, cnt;

    // inserimento delle dimensioni effettive
    do
    {
        printf("\n inserire numero di righe");
        scanf("%d", &r);
    }while(r < 1 || r > N);

    do
    {
        printf("\n inserire numero di colonne");
        scanf("%d", &c);
    }while(c < 1 || c > N);

    // inserimento della matrice
    for(i = 0; i < r; i++)
    {
        for(j = 0; j < c; j++)
        {
            printf("\nM[%d][%d] = ", i, j);
            scanf("%d", &M[i][j]);
        }
    }

    // stampa la matrice -- non richiesto.
    for(i = 0; i < r; i++)
    {
        for(j = 0; j < c; j++)
            printf("%5d", M[i][j]);
        printf("\n");
    }

    // scorre la matrice per colonne e copia tutti i numeri pari in un vettore vet, fino al
    primo multiplo di 7

    cnt = 0;
    flag = 1;
    for(j = 0; j < c && flag; j++)
        for(i = 0; i < r && flag; i++)
        {
            if(M[i][j] % 2 == 0)
            {
                vet[cnt] = M[i][j];
                cnt++;
            }
            if(cnt >= r)
                flag = 0;
        }

    // stampa il vettore
```

```

printf("\n[");
for(i = 0; i < cnt; i++)
    printf("%d, ", vet[i]);
printf("]\n");

printf("\n La matrice");
if(flag)
    printf(" NON");
printf(" ha almeno %d elementi pari\n",r);
}

```

TEMA B Esercizio 1 (4 punti)

Si consideri la seguente espressione booleana:

$$((\text{NOT } B) \text{ OR } A) \text{ OR } ((\text{NOT } A) \text{ AND } B) \text{ AND } C$$

2. Si compili la seguente **tabella della verità** (in cui 0 rappresenta il valore logico FALSO, 1 il valore VERO):

A	B	C	NOT B	(NOT B) OR A	NOT A	(NOT A) AND B	((NOT B) OR A) OR ((NOT A) AND B)	((NOT B) OR A) OR ((NOT A) AND B) AND C
0	0	0						
0	0	1						
0	1	0						
0	1	1						
1	0	0						
1	0	1						
1	1	0						
1	1	1						

3. Si consideri ora la condizione, scritta in linguaggio C, in cui **x** e **y** siano due variabili int:

$$((!(y>0) \text{ || } x>3) \text{ || } (!(x>3) \text{ \&\& } y>0)) \text{ \&\& } x<0$$

ottenuta dall'espressione booleana sopra indicata sostituendo la variabile A con **x>3**, la variabile B con **y>0**, la variabile C con **x<0**

Si risponda alle seguenti domande:

- c. L'espressione e' vera o falsa quando x=-5 e y=7? (giustificare la risposta)
- d. Se x<0, per quali valori di y l'espressione e' vera? (giustificare la risposta)

Soluzione

3. L'intera espressione equivale a C

A	B	C	NOT B	(NOT B) OR A	NOT A	(NOT A) AND B	((NOT B) OR A) OR ((NOT A) AND B)	((NOT B) OR A) OR ((NOT A) AND B) AND C
0	0	0	1	1	1	0	1	0
0	0	1	1	1	1	0	1	1
0	1	0	0	0	1	1	1	0
0	1	1	0	0	1	1	1	1
1	0	0	1	1	0	0	1	0
1	0	1	1	1	0	0	1	1
1	1	0	0	1	0	0	1	0
1	1	1	0	1	0	0	1	1

4. A) Quando $x = -5 \rightarrow C = 1 \rightarrow$ l'espressione è vera
B) Se $x < 0 \rightarrow C = 1 \rightarrow$ l'espressione è vera per qualsiasi valore di y

Esercizio 2 (7 punti)

Si considerino le seguenti strutture dati, che permettono di memorizzare le informazioni di 20 piloti che partecipano ad una gara automobilistica di 50 giri. In particolare, vengono memorizzati il nome del pilota, il nome della sua squadra, il numero del pilota in gara, il numero di giri completati ed il tempo realizzato in ciascun giro completato.

```
typedef struct
{
    int minuti;
    int secondi;
    int millesimi;
} tempo;

typedef struct
{
    char nome[50];
    char squadra[50];
    int numero;
    int giri_completati;
    tempo giri[50];
} pilota;
```



```
pilota piloti[20];
```

- D. Si modifichino opportunamente le strutture dati qui sopra per memorizzare, per ogni pilota, anche:
- il tempo totale di gara realizzato (cioè il tempo complessivo impiegato per completare tutti i giri effettuati);
 - se il pilota ha effettivamente completato i 50 giri previsti dalla gara.
- E. Scrivere un frammento di programma C che, per ciascun pilota, calcoli le informazioni aggiuntive indicate nel punto precedente (tempo totale e l'informazione sul completamento della gara).
- F. Scrivere un frammento di programma C che stampi a video nome, squadra e numero del pilota che (nel corso della gara) ha completato i 50 giri della gara nel minor tempo. Se nessun pilota ha completato tutti i giri, il frammento di programma dovrà stampare a video il messaggio "Gara ancora in corso, si prega di attendere".

Note: Si assuma che la variabile `piloti` sia già stata inizializzata (non è perciò richiesto di inizializzarla o acquisirla da tastiera). Per semplificare le operazioni richieste al punto B e C, si suggerisce di rappresentare il tempo totale in secondi. Ad esempio, un tempo di 20 minuti, 40 secondi e 123 millesimi, può essere rappresentato con il suo valore in secondi: 1240.123

Soluzione

1)

```
typedef struct
{
    char nome[50];
    char squadra[50];
    int numero;
    int giri_completati;
    tempo giri[50];
    float totale;
    int ha_finito;
} pilota;
```

2)

```
int i, j;

for (i=0; i<20; i++)
{
    piloti[i].ha_finito = piloti[i].giri_completati == 50;
    piloti[i].totale = 0;
    for (j=0; j<piloti[i].giri_completati; j++)
        piloti[i].totale = piloti[i].giri[j].minuti*60+
            piloti[i].giri[j].secondi +
            piloti[i].giri[j].millesimi/1000.0;
}
```

3)

```

int ii=0;

for (i=0; i<20; i++)
{
    if ( (piloti[i].totale < piloti[ii].totale && piloti[i].ha_finito ==
1))
        ii = i;
}
if (piloti[ii].ha_finito == 1)
printf("Il tempo di gara migliore è stato realizzato da %s, numero %d,
squadra
    %s\n",piloti[ii].nome,piloti[ii].numero,piloti[ii].squadra);
else printf("Gara ancora in corso, si prega di attendere\n");

```

Esercizio 3 (6 punti)

Si scriva un programma C per analizzare una matrice di interi M . La matrice deve essere definita con dimensioni 50×50 . Il programma dovrà:

5. Richiedere all'utente di inserire il numero di righe r e il numero di colonne c che desidera inserire, controllando che non superino le dimensioni massime consentite.
6. Richiedere l'inserimento di tutti gli elementi della matrice.
7. Scorrere la matrice per colonne e copiare in un vettore v tutti gli elementi *dispari* fino a un massimo di c elementi.
8. Stampare il contenuto di v a schermo e dire se la matrice contiene almeno c elementi *dispari*.

N.B. Scorrere "la matrice per colonne" vuol dire analizzare interamente la prima colonna della matrice dal primo all'ultimo elemento, quindi scorrere allo stesso modo la seconda colonna e così via fino all'ultima colonna.

Esempio

Se $M =$

2	23	7	2
4	2	5	1
31	5	8	5

il programma stamperà a schermo

$v = [31, 23, 5, 7]$

La matrice ha almeno 4 elementi dispari

Soluzione

```

#include <stdio.h>
#define N 50

void main()
{
    int M[N][N];
    int vet[N];
    int r,c,i,j,cnt, flag;

    // inserimento delle dimensioni effettive
    do
    {

```

```

        printf("\n inserire numero di righe");
        scanf("%d", &r);
}while(r < 1 || r > N );

do
{
    printf("\n inserire numero di colonne");
    scanf("%d", &c);
}while(c < 1 || c > N);

// inserimento della matrice
for(i = 0; i < r; i++)
{
    for(j = 0; j < c; j++)
    {
        printf("\nM[%d][%d] = ", i, j);
        scanf("%d", &M[i][j]);
    }
}

// stampa la matrice -- non richiesto.
for(i = 0; i < r; i++)
{
    for(j = 0; j < c; j++)
        printf("%5d", M[i][j]);
    printf("\n");
}

// scorre la matrice per colonne
// copia tutti i numeri maggiori del numero di colonne in un vettore fino ad un massimo
di c elementi
cnt = 0;
flag = 1;
for(j = 0; j < c && flag; j++)
    for(i = 0; i < r && flag; i++)
    {
        if(M[i][j] % 2 == 1)
        {
            vet[cnt] = M[i][j];
            cnt++;
        }
        if(cnt >= c)
            flag = 0;
    }

printf("\n La matrice");
if(flag)
    printf(" NON");
printf(" ha almeno %d elementi dispari\n",c);

// stampa il vettore
printf("\n[");
for(i = 0; i < cnt; i++)
    printf("%d, ", vet[i]);
printf("]\n");
}

```

TEMA C Esercizio 1 (4 punti)

Si consideri la seguente espressione booleana:

$((\text{NOT } C) \text{ AND } A) \text{ AND } (C \text{ OR } (\text{NOT } A)) \text{ OR } B$

3. Si compili la seguente **tabella della verità** (in cui 0 rappresenta il valore logico FALSO, 1 il valore VERO):

A	B	C	NOT C	(NOT C) AND A	NOT A	C OR (NOT A)	((NOT C) AND A) AND (C OR (NOT A))	((NOT C) AND A) AND (C OR (NOT A)) OR B
0	0	0						
0	0	1						
0	1	0						
0	1	1						
1	0	0						
1	0	1						
1	1	0						
1	1	1						

4. Si consideri ora la condizione, scritta in linguaggio C, in cui **x** e **y** siano due variabili int:

$((!(y>3) \ \&\& \ x<0) \ \&\& \ (y>3 \ || \ !(x<0))) \ || \ x>0$

ottenuta dall'espressione booleana indicata, sostituendo la variabile A con **x<0**, la variabile B con **x>0**, la variabile C con **y>3**

Si risponda alle seguenti domande:

- e. L'espressione e' vera o falsa quando x=5 e y=7? (giustificare la risposta)
- f. Se x=0, per quali valori di y l'espressione e' vera? (giustificare la risposta)

Soluzione

5. L'intera espressione equivale a B

A	B	C	NOT C	(NOT C) AND A	NOT A	C OR (NOT A)	((NOT C) AND A) AND (C OR (NOT A))	((NOT C) AND A) AND (C OR (NOT A)) OR B
0	0	0	1	0	1	1	0	0
0	0	1	0	0	1	1	0	0
0	1	0	1	0	1	1	0	1
0	1	1	0	0	1	1	0	1
1	0	0	1	1	0	0	0	0
1	0	1	0	0	0	1	0	0
1	1	0	1	1	0	0	0	1
1	1	1	0	0	0	1	0	1

6. A) Quando $x=5 \rightarrow B=1 \rightarrow$ l'espressione è vera

B) Se $x=0 \rightarrow B=0 \rightarrow$ l'espressione è falsa per qualsiasi valore di y

Esercizio 2 (7 punti)

Si considerino le seguenti strutture dati, che permettono di memorizzare le informazioni di 20 piloti che partecipano ad una gara automobilistica di 50 giri. In particolare, vengono memorizzati il nome del pilota, il nome della sua squadra, il numero del pilota in gara, il numero di giri completati ed il tempo realizzato in ciascun giro completato.

```
typedef struct
{
    int minuti;
    int secondi;
    int millesimi;
} tempo;
```

```
typedef struct
{
    char nome[50];
    char squadra[50];
    int numero;
    int giri_completati;
    tempo giri[50];
} pilota;
```

```
pilota piloti[20];
```

- G. Si modifichino opportunamente le strutture dati qui sopra per consentire di memorizzare, per ogni pilota, anche:
- il tempo medio sul giro (cioè il tempo medio impiegato a compiere un giro) effettuato da ciascun pilota nel corso della gara;
 - se il pilota ha effettivamente completato i 50 giri previsti dalla gara.
- H. Scrivere un frammento di programma C che, per ciascun pilota, calcoli le informazioni aggiuntive indicate nel punto precedente (tempo medio sul giro e l'informazione sul completamento della gara).
- I. Scrivere un frammento di programma C che stampi a video nome, squadra e numero del pilota che (nel corso della gara) ha completato i 50 giri previsti dalla gara con il maggior tempo medio sul giro. Se nessun pilota ha completato tutti i giri, il frammento di programma dovrà stampare a video il messaggio "Gara ancora in corso, si prega di attendere".
- 4.
5. **Note:** Si assuma che la variabile `piloti` sia già stata inizializzata (non è perciò richiesto di inizializzarla o acquisirla da tastiera). Per semplificare le operazioni richieste al punto B e C, si suggerisce di rappresentare il tempo medio sul giro in secondi. Ad esempio, un tempo di 2 minuti, 4 secondi e 123 millesimi, può essere rappresentato con il suo valore in secondi: 124.123

Soluzione

1)

```
typedef struct
{
    char nome[50];
    char squadra[50];
    int numero;
    int giri_completati;
    tempo giri[50];
    float media;
    int ha_finito;
} pilota;
```

2)

```
int i, j, t;
float somma;

for (i=0; i<20; i++)
{
    piloti[i].ha_finito = piloti[i].giri_completati == 50;
    somma = 0;
    for (j=1; j<piloti[i].giri_completati; j++)
    {
        media += piloti[i].giri[j].minuti*60+piloti[i].giri[j].secondi +
                piloti[i].giri[j].millesimi/1000.0;
    }
    piloti[i].media = somma/piloti[i].giri_completati;
}
```

3)

```
int ii=0;

for (i=0; i<20; i++)
{
    if (piloti[i].media > piloti[ii].media && piloti[i].ha_finito == 1)
        ii = i;
}
if (piloti[ii].ha_finito == 1)
printf("Il tempo medio maggiore è stato realizzato da %s, numero %d,
squadra
%s\n",piloti[ii].nome,piloti[ii].numero,piloti[ii].squadra);
else printf("Gara ancora in corso, si prega di attendere\n");
```

Esercizio 3 (6 punti)

Si scriva un programma C per analizzare una matrice di interi M . La matrice deve essere definita con dimensioni 50×50 . Il programma dovrà:

9. Richiedere all'utente di inserire il numero di righe r e il numero di colonne c che desidera inserire, controllando che non superino le dimensioni massime consentite.
 10. Richiedere l'inserimento di tutti gli elementi della matrice.
 11. Scorrere la matrice per righe e copiare in un vettore v tutti i numeri dispari fino ad un massimo di c elementi.
 12. Stampare il contenuto di v a schermo e dire se la matrice contiene almeno c elementi dispari.
6.
N.B. Scorrere "la matrice per righe" vuol dire analizzare interamente la prima riga della matrice dal primo all'ultimo elemento, quindi scorrere allo stesso modo la seconda riga e così via fino all'ultima riga.

Esempio

Se $M =$

12	3	1	2
8	4	7	22

il programma stamperà a schermo

$v = [3, 1, 7]$

La matrice NON contiene almeno 4 elementi dispari.

Soluzione

```
#include <stdio.h>
#define N 10

void main()
{
    int M[N][N];
    int vet[N*N];
    int r,c,i,j,flag, cnt;
```

```

// inserimento delle dimensioni effettive
do
{
    printf("\n inserire numero di righe");
    scanf("%d", &r);
}while(r < 1 || r > N);

do
{
    printf("\n inserire numero di colonne");
    scanf("%d", &c);
}while(c < 1 || c > N);

// inserimento della matrice
for(i = 0; i < r; i++)
{
    for(j = 0; j < c; j++)
    {
        printf("\nM[%d][%d] = ", i, j);
        scanf("%d", &M[i][j]);
    }
}

// stampa la matrice -- non richiesto.
for(i = 0; i < r; i++)
{
    for(j = 0; j < c; j++)
        printf("%5d", M[i][j]);
    printf("\n");
}

// scorre la matrice per colonne e copia tutti i numeri pari in un vettore vet, fino al
primo multiplo di 7

cnt = 0;
flag = 1;
for(j = 0; j < c && flag; j++)
    for(i = 0; i < r && flag; i++)
    {
        if(M[i][j] % 2 == 1)
        {
            vet[cnt] = M[i][j];
            cnt++;
        }
        if(cnt >= c)
            flag = 0;
    }

// stampa il vettore
printf("\n[");
for(i = 0; i < cnt; i++)
    printf("%d, ", vet[i]);
printf("]\n");

printf("\n La matrice");
if(flag)
    printf(" NON");
printf(" ha almeno %d elementi dispari\n",r);
}

```