

	Politecnico di Milano Facoltà di Ingegneria Industriale INFORMATICA B Prova in itinere del 18 Novembre 2008		COGNOME E NOME
	RIGA	COLONNA	MATRICOLA

- Il presente plico contiene 4 esercizi, deve essere debitamente compilato con cognome e nome, numero di matricola, posizione durante lo scritto (comunicata dal docente).
- Il tempo a disposizione è di 2 ore.
- 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!).
- 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.
- 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 (3 punti)

Per ognuna delle espressioni logiche riportate nelle righe della tabella, assumendo le seguenti dichiarazioni:

```
int a = 8, b = 11;
char c='d';
```

indicare se l'espressione è vera o falsa (scrivere V o F nella seconda colonna). Indicare inoltre, nella terza colonna, se l'espressione è vera per qualsiasi valore delle variabili (scrivere SI o NO) e, nella quarta colonna, se l'espressione è falsa per qualsiasi valore delle variabili (scrivere SI o NO). Si giustificino le risposte. Risposte prive di giustificazione non saranno prese in considerazione.

Espressione	Vera o falsa?	Sempre vera?	Sempre falsa?
<code>(-a==a) && (a < 11)</code>	F	NO	NO
<code>(c>'a' c<'z') && (a<7 && b>8)</code>	F	NO	NO
<code>!(b<10 && a>7) (c!='h' && c>'a')</code>	V	NO	NO

Giustificazioni

1. L'espressione è vera nel caso $a=0$ e falsa in tutti gli altri casi
2. L'espressione è falsa per i valori dati per la presenza del termine $a<7$; è vera, per valori di $a<7$ e $b>8$ perché il termine $(c>'a' || c<'z')$ è sempre vero. Di conseguenza, l'espressione non è sempre falsa.
3. L'espressione è vera perché, indipendentemente dai valori di a e b , si ha che $c!='h'$ e $c>'a'$ sono entrambe vere; è falsa, per esempio, per $a=8$, $b=9$ (che rendono falso il termine $!(b<10 \ \&\& \ a>7)$) e $c='h'$ (che rende falso il termine $(c!='h' \ \&\& \ c>'a')$), quindi non è sempre vera.

Esercizio 2 (5 punti)

Le seguenti dichiarazioni definiscono tipi di dati che descrivono i candidati per l'assunzione in un'azienda.

```
/* definizioni dei tipi */
typedef enum {italiano, inglese, spagnolo, francese, tedesco, cinese, portoghese} lingua;
typedef char stringa[20];
typedef struct {
    lingua l;
    int livello; /* numero da 1 a 5 con 1=basso 5=alto */
} linguaParlata;
typedef struct {
    stringa nome, cognome, diploma, laurea;
    linguaParlata lingue[5];
    int nLauree;
    int anniEsperienza;
} persona;

/* definizioni delle variabili */
persona persone[40];
persona personeScelte[40];
```

Si scriva un frammento di codice, che includa eventualmente anche le dichiarazioni di ulteriori variabili e tipi, che copi nella parte iniziale del vettore personeScelte (senza lasciare buchi) le persone che parlano inglese con un livello superiore a 3 oppure soddisfano entrambi i seguenti requisiti: hanno non meno di una laurea e un numero di anni di esperienza non inferiore a tre.

```
int iPersone, iPersoneScel, iLingue;
iPersoneScel=0;
for(iPersone=0; iPersone<40; iPersone++) {
    if (persone[iPersone].nLauree>=1 && persone[iPersone].anniEsperienza>=3){
        personeScelte[iPersoneScel]=persone[iPersone];
        iPersoneScel++;
    }else for (iLingue=0; iLingue<5; iLingue++)
        if(persone[iPersone].lingue[iLingue].l==inglese &&
            persone[iPersone].lingue[iLingue].livello >3) {
            personeScelte[iPersoneScel]=persone[iPersone];
            iPersoneScel++;
        }
}
```

Esercizio 3 (4 punti)

Si consideri il seguente programma C

```
#include <stdio.h>
#define MAX 20

int main()
{
    /* la definizione del tipo puo` trovarsi anche fuori dal main */
    typedef int numeri[MAX];

    numeri A;
    int p,j,i=0;
    int s=0;

    do{
        scanf("%d",&A[i]);
        i++;
    }while(i<MAX && A[i-1]>0);
    do{
        scanf("%d",&p);
    }while(p<0 || p>=i);

    for(j=0;j<p;j++)
        s=s+A[j];

    for(j=p;j<i;j++)
        s=s-A[j];

    if(s==0)
        printf("%d",p);

    else if(s<0)
        for(j=p;j<i;j++)
            printf("%d\n",A[j]);

    else
        for(j=0;j<p;j++)
            printf("%d\n",A[j]);
}
```

1. Si indichi cosa viene stampato a video se l'utente inserisce questa sequenza:
4 5 12 12 3 3 3 3 7 14 0 4
2. Che cosa succede se l'utente invece di scrivere 4 come ultimo valore della sequenza scrive 3? Che cosa succede se invece scrive 12?
3. Spiegare brevemente il funzionamento del programma a seconda dei vari possibili ingressi

Soluzione

1. Il programma legge da tastiera una sequenza di numeri fino ad incontrare il valore 0. Poi legge un ulteriore valore (nell'esempio il valore 4) che viene usato per dividere la sequenza di numeri letta in precedenza in due porzioni distinte (nell'esempio, i primi 4 numeri della sequenza e gli altri). A questo punto il programma calcola la differenza tra la somma dei numeri della prima porzione della sequenza e la somma dei numeri della seconda porzione della sequenza. Se tale differenza è pari a zero, il programma stampa il valore che è stato utilizzato per dividere la sequenza in due parti (il valore 4 nel nostro esempio). Se la differenza è minore di zero il programma stampa tutti i valori nella seconda porzione della sequenza. Se è maggiore di zero, il programma stampa tutti i valori nella prima porzione della sequenza. Nel caso dell'esempio specifico, la differenza è pari a zero e, di conseguenza, viene stampato il valore 4.
2. Nel caso in cui l'utente inserisce 3 come ultimo elemento, il programma divide la sequenza nelle due sottosequenze composte dai numeri {4, 5, 12} la prima e dai numeri {12, 3, 3, 3, 3, 7, 14, 0} la seconda. In questo caso quindi la differenza calcolata risulta pari a -24. Di conseguenza, il programma stamperà a video i numeri 12 3 3 3 3 7 14 0.
Nel caso in cui l'utente inserisce come ultimo valore il numero 12, la condizione del ciclo che si occupa dell'acquisizione dell'ultimo valore risulterà essere vera perché il termine $p \geq i$ sarà vero. Di conseguenza, il programma, rimarrà in attesa di ricevere un nuovo valore.
3. Si veda la spiegazione al punto 1.

Esercizio 4 (5 punti)

Si scriva un programma C che acquisisce una matrice di interi di nome `matr` e di dimensione $N \times N$ (con N definito come costante) e due interi X e Y da standard input.

Se X e Y non sono indici ammissibili della matrice, il programma termina. In caso contrario, il programma stampa la stringa "successo!" se l'elemento `matr[X][Y]` è uguale a $X*Y$. Se quest'ultima condizione non è verificata, il programma considera gli elementi della riga X in `matr` e controlla se tra questi quelli maggiori di `matr[X][Y]` sono di più di quelli minori di `matr[X][Y]`. Se questo è il caso, il programma stampa il valore contenuto in `matr[X][Y]`.

Ad esempio, se la riga X di `matr` è costituita dagli elementi $\{12, 7, 15, 5, 3\}$ e l'elemento che stiamo considerando è quello di posizione 1 (il valore 7), possiamo concludere che due elementi della riga sono minori di tale numero e altri due sono maggiori di esso. Di conseguenza, la condizione non è verificata e quindi il programma non stampa nulla.

Soluzione

```
#include <stdio.h>
#define N 5

void main()
{
    int matr[N][N];
    int X, Y, i, j, contMin, contMagg;

    for(i=0; i<N; i++)
        for(j=0; j<N; j++)
            scanf("%d", &matr[i][j]);
    scanf("%d", &X);
    scanf("%d", &Y);
    contMin=0;
    contMagg=0;
    if(X>=0 && X<N && Y>=0 && Y<N)
    {
        if(matr[X][Y] == X*Y) printf("successo!\n");
        else
        {
            for(j=0; j<N; j++)
            {
                if(matr[X][j] < matr[X][Y]) contMin++;
                if(matr[X][j] > matr[X][Y]) contMagg++;
            }
            if(contMagg>contMin)
                printf("%d\n", matr[X][Y]);
        }
    }
}
```