

UNIVERSITÀ DI UDINE  
CDL IN MATEMATICA  
PROVA SCRITTA DI INFORMATICA 1  
12 Giugno 2002

1. (5pt) Si descriva il ciclo principale della CPU di un calcolatore e si fornisca una classificazione delle tipologie di operazioni elementari che esso può eseguire.
2. (5pt) Si enunci e dimostri il Teorema di Böhm-Jacopini.
3. (5pt) Il testo CIAO viene codificato usando il codice ASCII (si ricorda che il codice di A è 65) con l'ottavo bit assegnato in modo tale da permettere il test di parità. Se leggiamo i 4 bytes come numero intero in C2, a quale numero ci si riferisce?
4. (5pt) Si spieghi qual è il significato del seguente frammento di codice Java (si assuma che le variabili in gioco siano già state opportunamente dichiarate) e se ne calcoli la complessità computazionale in funzione del valore di  $n$ :

```
n = MioIO.LeggiIntero();
k = 0;
for(i = 1; i <= n ; i = i + 1)
    for(j = i + 1; j <= n; j = j + 1)
        k = k + i + j;
MioIO.Scrivi(k);
```

5. (10pt) Si scriva un metodo Java che:
  - accetta in input due valori di tipo carattere  $x$  e  $y$  e una matrice  $m$  di dimensione  $n \times 2$  di caratteri.
  - restituisce l'indice più basso della riga in cui la coppia  $\langle x, y \rangle$  viene trovata.

Si calcoli la complessità di tale metodo in funzione di  $n$ . Si mostri come tale metodo possa essere richiamato dal `main` o da altri metodi.

Si assuma ora che la matrice abbia le righe ordinate secondo l'ordine lessicografico degli elementi in esse contenuti, ovvero:

$$i < j \rightarrow \left( \begin{array}{l} m[i][0] < m[j][0] \vee \\ m[i][0] = m[j][0] \wedge m[i][1] \leq m[j][1] \end{array} \right)$$

Si scriva un secondo metodo Java che ottimizza il precedente nel caso la condizione sopra sia verificata e se ne calcoli la complessità.

UNIVERSITÀ DI UDINE  
CDL IN MATEMATICA  
PROVA SCRITTA DI INFORMATICA 1  
27 Giugno 2002

1. (5pt) Si descriva la memoria cache: il suo scopo e la metodologia di accesso per le operazioni di lettura e scrittura.
2. (5pt) Si illustri come un frammento di diagramma di flusso rappresentante un generico costrutto `if / then / else` possa essere simulato mediante un frammento di diagramma di flusso fortemente strutturato.
3. (5pt) Si rappresenti, con la notazione in virgola mobile IEEE su 32 bit, il numero 1936,27.
4. (7pt) Si spieghi qual è l'output del seguente programma Java in funzione del valore `n` letto dall'input. Se ne calcoli dunque la complessità computazionale in funzione del valore di `n`:

```
public class EsameMat {
    public static void a(int x, int [] v){
        b(x, x, v);
        x = x - 1;
    }
    public static void b(int y, int z, int [] v){
        if (y > 0) {
            v[0] = v[0] + 1;
            b(y-1, z, v);
        }
        else if (z > 0)
            a(z-1,v);
    }
    public static void main( String [] args){
        int n;
        int [] v = new int [1];
        v[0] = 0;
        n = MioIO.LeggiIntero("Dammi un numero");
        a(n,v);
        MioIO.Scrivi(n + " " + v[0]);
        MioIO.Finisci();
    }
}
```

5. (8pt) Si scriva un *metodo* Java che:
  - accetta in input due valori interi ( $x$  e  $y$ ) e
  - *restituisce* il loro minimo comune multiplo.

Non è richiesto di scrivere il programma principale!!! Se invece sono necessari altri metodi ausiliari, li si definisca. Si dimostri la correttezza del metodo e si calcoli la complessità in funzione di  $h = \max\{|x|, |y|\}$ .

UNIVERSITÀ DI UDINE  
CDL IN MATEMATICA  
PROVA SCRITTA DI INFORMATICA 1  
17 Luglio 2002

1. (5pt) Si descriva brevemente la memoria principale del calcolatore. In particolare si mostri a composizionalità della stessa (ovvero come si possano combinare più chip di memoria per ottenerne una di capacità maggiore).
2. (5pt) Si illustri mediante un diagramma di flusso l'algoritmo che calcola la parte intera del logaritmo in base 2 di un numero naturale positivo dato.
3. (5pt) Si effettui la somma dei due numeri -41 e -120 dopo averli rappresentati in complemento a 2 su 8 bit. Si discuta il risultato ottenuto.
4. (5pt) Si spieghi qual è l'output del seguente programma Java in funzione dei valori  $m$ ,  $n$  letto dall'input. Se ne calcoli dunque la complessità computazionale in funzione dei valori di  $m$ ,  $n$ :

```
public class EsameMat {
    public static int a(int x, int y){
        return x * y;
    }
    public static void main( String [] args){
        int m,n,i,j;
        m = MioIO.LeggiIntero("Dammi un numero");
        n = MioIO.LeggiIntero("Dammi un numero");
        j = 0;
        for (i = 1; i <= a(m,n); i = i + 1)
            j = j + a(i,i);
        MioIO.Scrivi(j);
        MioIO.Finisci();
    }
}
```

5. (10pt) Si scrivano i seguenti *metodi* Java:
  - (a) il metodo `check` che stabilisce se il vettore di byte che gli viene passato come argomento contiene solo numeri dallo 0 al 9.
  - (b) il metodo `sum` che accetta come input due vettori di byte: se entrambi i vettori passano il test di `check`, allora fornisce come output il vettore di byte che corrisponde alla loro somma (i due vettori possono avere dimensioni diverse); altrimenti restituisce un vettore di -1.

Non è richiesto di scrivere il programma principale!!! Se invece fossero necessari altri metodi ausiliari non standard, li si definisca. Si calcoli la complessità dei metodi descritti in funzione delle dimensioni dei vettori passati come parametri.

UNIVERSITÀ DI UDINE  
CDL IN MATEMATICA  
PROVA SCRITTA DI INFORMATICA 1  
5 Settembre 2002

1. Si descriva mediante un diagramma di flusso il ciclo principale della CPU di un calcolatore.
2. Si spieghi inoltre entrando in dettaglio le varie operazioni compiute dalle CPU per effettuare una somma di due valori (presenti in memoria) in ASSEMBLER.
3. Si effettui la seguente operazione su aritmetica binaria in complemento a 2 su 16 bit:  $1000 - 2000$ .
4. Si spieghi qual è l'output del seguente programma Java in funzione del valore  $n$  letto dall'input. Se ne calcoli dunque la complessità computazionale in funzione del valore di  $n$ :

```
public class EsameMat {
    public static void a(int x, int [] v){
        if (x > 0){
            v[0] = v[0] + x;
            a(x-1, v);
        }
        else x = x -1;
    }
    public static void main( String [] args){
        int n;
        int [] v = new int [1];
        v[0] = 0;
        n = MioIO.LeggiIntero("Dammi un numero");
        a(n,v);
        MioIO.Scrivi(n + " " + v[0]);
        MioIO.Finisci();
    }
}
```

5. Si scrivano i seguenti *metodi* Java:
  - (a) il metodo **check** che stabilisce se il vettore di byte che gli viene passato come argomento contiene solo numeri dallo 0 al 9.
  - (b) il metodo **diff** che accetta come input due vettori di byte: se entrambi i vettori passano il test di **check**, allora fornisce come output il vettore di byte che corrisponde alla loro differenza (i due vettori possono avere dimensioni diverse); altrimenti restituisce un vettore di -1.

Non è richiesto di scrivere il programma principale!!! Se invece fossero necessari altri metodi ausiliari non standard, li si definisca. Si calcoli la complessità dei metodi descritti in funzione delle dimensioni dei vettori passati come parametri.

UNIVERSITÀ DI UDINE  
CDL IN MATEMATICA  
PROVA SCRITTA DI INFORMATICA 1  
24 Settembre 2002

1. Si descriva l'organizzazione fisica e logica di un supporto magnetico di memoria secondaria.
2. Si descriva, mediante un diagramma di flusso, *un* algoritmo di ordinamento di un vettore.
3. Si effettui la seguente operazione in aritmetica binaria in complemento a 2 su 8 bit: -128 -1.
4. Si spieghi qual è l'output del seguente programma Java in funzione del valore *n* letto dall'input. Se ne calcoli dunque la complessità computazionale in funzione del valore di *n*:

```
public class EsameMat {
    public static void a(int n, int [] v){
        if (n > 0){
            v[0] = v[0] * n;
            a(n-1, v);
        }
        else n = 5;
    }
    public static void main( String [] args){
        int n;
        int [] v = new int [1];
        v[0] = 1;
        n = MioIO.LeggiIntero("Dammi un numero");
        a(n,v);
        MioIO.Scrivi(n + " " + v[0]);
        MioIO.Finisci();
    }
}
```

5. Si scriva un *metodo* Java che accetta come input due vettori di byte. Assumiamo che tali vettori contengano solo numeri dallo 0 al 9; ciascun vettore nel suo complesso rappresenta un numero intero rappresentato nel sistema decimale. Il metodo deve restituire (non stampare!) un vettore che rappresenta il prodotto di tali numeri interi.

Suggerimento: Si scriva prima un metodo in cui si assume che il secondo vettore abbia esattamente due cifre, con eventualmente la prima a 0. Solo in un secondo tempo, se rimane tempo e dopo averlo spiegato bene e studiato la complessità (vedi sotto) si passi al metodo generale.

Si calcoli la complessità del metodo/metodi descritti in funzione delle dimensioni dei vettori passati come parametri.

UNIVERSITÀ DI UDINE  
CDL IN MATEMATICA  
PROVA SCRITTA DI INFORMATICA 1  
17 Giugno 2003

1. (5pt) Si descriva il ciclo principale della CPU di un calcolatore.
2. (5pt) Si dimostri che, dato un linguaggio di programmazione, esistono funzioni in  $\{f : \mathbb{N} \rightarrow \mathbb{N}\}$  che non possono essere calcolate.
3. (5pt) Si esegua, in Complemento a 2 su 16 bit, l'operazione  $30000 + 20000$ . Quale risultato sarebbe stampato?
4. (6pt) Si spieghi qual è il significato del seguente codice Java e se ne calcoli la complessità computazionale in funzione del valore di  $n$ . In particolare, qual è il valore finale di  $v[0]$  ?

```
public class Esame1 {
public static void reset(int [] v){
    int i;
    for(i = 0; i < v.length; i = i + 1)
        v[i] = 0;
}
public static void stampa(int [] v){
    int i;
    for(i = 0; i < v.length; i = i + 1)
        System.out.print(v[i] + " ");
}
public static void p(int a,int b, int [] v){
    int i;
    if (a < b){
        for(i = a; i <= b; i = i + 1)
            v[i] = v[i] + 1;
        p(a, (a + b) / 2, v);
    }
}
public static void main (String [] args) {
    int [] w;
    int n;

    n = MioIO.LeggiIntero();
    w = new int [n];
    reset(w);
    p(0,w.length - 1, w);
    stampa(w);

    MioIO.Finisci();
}
```

5. (9pt) Si scriva un metodo Java che: accetta in input 3 numeri naturali  $x, y, z$  rappresentati mediante vettori di bytes e restituisce in output **true** se  $z$  passa il test della *prova del nove* nella verifica che  $z = x \cdot y$ , **false** altrimenti. Si discuta della complessità del metodo sviluppato.

UNIVERSITÀ DI UDINE  
CDL IN MATEMATICA  
PROVA SCRITTA DI INFORMATICA 1  
4 Luglio 2003

1. (5pt) Si elenchino le tipologie di memoria secondaria attualmente utilizzate, e si discuta dei tempi di accesso e dell'organizzazione dei dischi magnetici.
2. (5pt) Si dimostri, mediante diagrammi di flusso, che il blocco `if ... then ... else` può essere *simulato* da una opportuna sequenza di blocchi `while`.
3. (5pt) Si consideri la sequenza di bytes necessari per rappresentare i caratteri A, B, C, D mediante il codice ASCII. Se letti come numero a virgola mobile a 32 bit, quale numero rappresentano?
4. (6pt) Si calcoli la complessità computazionale del metodo `m` sotto definito in funzione della dimensione del vettore `v`, suo parametro formale.

```
public static int r(int [] w){
    int m;
    m = w[0];
    for(i = 1; i < w.length; i = i + 2)
        m = (m * w[i]) % 5;
}
```

```
public static void n(int a, int b, int [] m){
    if (a < b)
        return r(m)*n(a+1,b-1,m);
}
```

```
public static int m (int [] v) {
    return n(0,v.length-1,v);
}
```

5. (9pt) Un grafo diretto  $G = \langle N, E \rangle$  si può rappresentare mediante una matrice quadrata `g` di booleani ove `g[i][j]` è `true` se e solo se esiste l'arco tra il nodo `i` ed il nodo `j` (si assuma  $N = \{0, 1, \dots, n\}$ ). Si scriva un metodo Java che: accetta in input, un grafo `g`, 2 numeri naturali `s` e `t` rappresentanti 2 nodi e ritorna `true` se esiste un cammino di lunghezza 2 (archi) tra `s` e `t`, `false` altrimenti. Si studi la complessità del metodo descritto.

Si generalizzi dunque il problema descrivendo un metodo che accetta un ulteriore parametro `l` e dice se esiste un cammino di lunghezza esattamente `l` tra `s` e `t`.

UNIVERSITÀ DI UDINE  
CDL IN MATEMATICA  
PROVA SCRITTA DI INFORMATICA 1  
22 Luglio 2003

1. (5pt) Si descrivano le modalità di scrittura e lettura di dati in memoria cache.
2. (5pt) Si descriva, mediante diagrammi di flusso, un algoritmo di ricerca binaria in un vettore ordinato.
3. (5pt) Si discuta della rappresentazione di immagini in bianco e nero, a livelli di grigio e in RGB nel calcolatore, con particolare riferimento alla quantità di memoria necessaria per immagini a  $1024 \times 768$  pixels.
4. (6pt) Si calcoli la complessità computazionale del metodo `m` sotto definito in funzione della dimensione del vettore `v`, suo parametro formale.

```
public static int r(int [] w) {  
    int i, m;  
    m = w[0] + w[1];  
    for (i = 2; i < w.length; i = i * i)  
        m = (m * w[i]) % 5;  
    return m;  
}
```

```
public static int n( int a, int b, int [] m){  
    if ( a < b)  
        return r(m)*n(a,(a+b)/2,m);  
    else return 1;  
}
```

```
public static int m (int [] v){  
    return n(0,v.length - 1,v);  
}
```

5. (9pt) Si scriva un metodo che prende come input un numero intero e fornisce come output un vettore booleano rappresentante la sua rappresentazione in complemento a 2 su 32 bits (si assuma `0 = false` e `1 = true`).

Si scriva dunque un metodo che prende come input due numeri rappresentati in complemento a 2 come nell'esercizio precedente e fornisce in output la loro somma. Nel caso vi siano situazioni di underflow o overflow, queste sono stampate (con `System.out.println()`).



UNIVERSITÀ DI UDINE  
CDL IN MATEMATICA  
PROVA SCRITTA DI INFORMATICA 1  
9 Settembre 2003

1. (5pt) Si descriva il ciclo principale della CPU di un calcolatore.
2. (5pt) Si illustri mediante un diagramma di flusso un algoritmo che calcola la parte intera della radice quadrata di un numero naturale positivo dato (le operazioni aritmetiche ammesse sono solo la somma/differenza e la moltiplicazione).
3. (5pt) Si esegua in complemento a 2 su 8 bits l'operazione  $-100 - 50$ . Si analizzi e discuta il risultato.

4. (6pt) Si studi la complessità in funzione di  $N$ , del seguente frammento di programma Java:

```
i = N;
j = N;
while (i > 0) {
    j = j + N;
    i = i - 1;
}
while (i < j) {
    i = i + 2;
}
```

5. (9pt) Si scriva un metodo che prende come input due vettori Booleani rappresentante la sua rappresentazione in complemento a 2 su 32 bits (si assuma  $0 = \text{false}$  e  $1 = \text{true}$ ) di due numeri interi e restituisce in output un vettore dello stesso tipo che rappresenta la loro differenza. Si studi la sua complessità. Nel caso vi siano situazioni di underflow o overflow, queste sono stampate (con `System.out.println()`).

UNIVERSITÀ DI UDINE  
CDL IN MATEMATICA  
PROVA SCRITTA DI INFORMATICA 1  
14 Giugno 2004

1. (5pt) Si descriva brevemente il ciclo di funzionamento della CPU accennando alle tipologie di istruzioni che quest'ultima è in grado di eseguire.
2. (5pt) Si illustri mediante diagramma di flusso un algoritmo per la conversione in binario di un numero naturale.
3. (6pt) Si dimostri che esiste una funzione da  $\mathbb{N}$  a  $\mathbb{N}$  che non possiamo codificare in C.
4. (6pt) Si spieghi qual è l'output del seguente programma C in funzione del valore  $n$  letto dall'input. Se ne calcoli dunque la complessità computazionale (approssimata) sempre in funzione del valore di  $n$ :

```
int a;
main(){
    int proc(int);
    int n;
    a = 1;
    scanf("%d",&n);
    printf("%d, %d, %d", a, n, proc(n));
}
int proc(int n){
    int i,j;
    for(i=1;i<=n;i++)
        for(j=i+1;j<=n;j++)
            a = a*2;
    n=a-2;
    return n;
}
```

5. (8pt) Si scriva una *funzione* C che:
  - accetta in input un valore intero  $x$  che sappiamo già essere compreso tra  $-256$  e  $+255$ .
  - *restituisce* come parametro un vettore Booleano di dimensioni opportune che rappresenta la rappresentazione in Complemento a due del numero  $x$ .

Non è richiesto di scrivere il programma principale, né di verificare le dimensioni di  $x$ !!! Se invece sono necessarie o comode delle funzioni ausiliarie, le si definisca.

Si calcoli la complessità della funzione in funzione del valore di  $x$ .

UNIVERSITÀ DI UDINE  
CDL IN MATEMATICA  
PROVA SCRITTA DI INFORMATICA 1  
30 Giugno 2004

1. (5pt) Si illustrino le architetture di Von Neumann e a Bus. Se la memoria è organizzata a parole di 8 Bytes e consta di 700 MBytes, come minimo di quanti bits sono costituiti il bus dati ed il bus indirizzi?
2. (5pt) Si supponga di disporre di Andy, un diplomato all'istituto d'arte e disoccupato, il quale è in grado di eseguire con precisione il comando `plot(x, y)` che stampa un punto nelle coordinate  $\langle x, y \rangle$  di un opportuno lenzuolo quadrato che funge da piano cartesiano. Egli può stampare con precisione di 0.01 punti con  $x$  e  $y$  a valori in  $\{-100.00, \dots, 100.00\}$ . Si scriva un algoritmo mediante diagrammi di flusso che permetta a Andy di stampare, senza uscire dal lenzuolo, la funzione  $2x^2 - 5x + 3$  nell'intervallo  $[-100, 100]$ .
3. (6pt) Si converta in decimale il numero floating point:

1	1	0	0	0	1	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

4. (6pt) Si spieghi qual è l'output del seguente programma C in funzione del valore  $n$  letto dall'input. Se ne calcoli dunque la complessità computazionale (approssimata) sempre in funzione del valore di  $n$ :

```
int a;
main(){
    int proca(int),procb(int);
    int n;
    a = 0;
    scanf("%d",&n);
    printf("%d, %d, %d", a, n, procb(proca(n)));
}
int proca(int n){
    int i,j;
    for(i=1;i<=n;i++)
        for(j=i+1;j<=n;j++)
            a = a+1;
    return n;
}
int procb(int n){
    int i;
    for(i=1;i<=a;i++)
        n = n-1;
    return n;
}
```

5. (8pt) Si scriva una *procedura* C `ordina(int v [DIM])` che ordina il vettore di DIM elementi che riceve in input. Come ulteriore ipotesi, sappiamo che tale vettore contiene solo numeri dallo 0 al 100: si cerchi di sfruttare questa ipotesi per scrivere una procedura che opera in tempo *lineare* su DIM. In ogni caso, si calcoli la complessità della procedura scritta in funzione del valore di DIM.

UNIVERSITÀ DI UDINE  
CDL IN MATEMATICA  
PROVA SCRITTA DI INFORMATICA 1  
2 Settembre 2004

1. (5pt) Si illustrino e descrivano in breve i principali supporti per la memoria secondaria.
2. (5pt) Si descriva, mediante diagrammi di flusso, una procedura di ordinamento di un vettore. Si supponga che in input si riceva la dimensione del vettore  $n$  e (tutto in una volta) il vettore  $v$ .
3. (6pt) Si converta in floating point su 32 bit il numero decimale 254,33.
4. (6pt) Si spieghi qual è l'output del seguente programma C in funzione del valore  $n$  letto dall'input. Se ne calcoli dunque la complessità computazionale (approssimata) sempre in funzione del valore di  $n$ :

```
int a;
main(){
    int proca(int),procb(int);
    int n;
    a = 0;
    scanf("%d",&n);
    printf("%d, %d, %d", a, n, proca(n));
}

int proca(int n){
    int procb (int);
    int i;
    for(i=1;i<=n;i++)
        a = a + procb(i);
    return n;
}

int procb(int n){
    int i,a;
    a = 0;
    for(i=1;i<=n;i++)
        a = a + i;
    return a;
}
```

5. (8pt) Un grafo diretto  $G = \langle N, E \rangle$  si può rappresentare mediante una matrice quadrata  $g$  di booleani ove  $g[i][j]$  è **true** se e solo se esiste l'arco tra il nodo  $i$  ed il nodo  $j$  (si assuma  $N = \{0, 1, \dots, n\}$ ). Si scriva una funzione Booleana  $C$  che: accetta in input un grafo  $g$ , 2 numeri naturali  $s$  e  $t$  rappresentanti 2 nodi e ritorna **true** se esiste un cammino di lunghezza al più 2 (archi) tra  $s$  e  $t$ , **false** altrimenti. Si studi la complessità della procedura descritta.

UNIVERSITÀ DI UDINE  
CDL IN MATEMATICA  
PROVA SCRITTA DI INFORMATICA 1  
22 Settembre 2004

1. (5pt) Si descriva brevemente il ciclo di funzionamento della CPU accennando alle tipologie di istruzioni che quest'ultima è in grado di eseguire.
2. (5pt) Si dimostri che esiste una funzione da  $\mathbb{N}$  a  $\mathbb{N}$  che non possiamo codificare in C.
3. (6pt) Si converta 125.125 in base 2. Successivamente lo si rappresenti nella codifica IEEE floating point su 32 bits.
4. (6pt) Si spieghi qual è l'output del seguente programma C in funzione del valore  $n$  letto dall'input. Se ne calcoli dunque la complessità computazionale.

```
int a;
main(){
    int proc(int);
    void procb(int);
    int n;
    a = 0;
    scanf("%d",&n);
    procb(n);
    printf("%d, %d, %d", n, proc(n), a);
}
int proc(int n){
    int i, a;
    a = 1;
    for(i=1;i<=n;i++)
        a = a*2;
    return n;
}
void procb(int n){
    int i;
    for(i=1;i<=n;i++)
        a = a+2;
}
```

5. (8pt) Si scriva una *funzione* C che:
  - accetta in input (come parametro) un vettore  $v$  di  $n$  elementi di tipo `float`, ed un intero  $m$ .
  - *restituisce* come parametro un vettore Booleano  $b$  di dimensione  $n$  tale che  $b[i]=\text{true}$  se e solo se  $v[i]>0$ .
  - *restituisce* come output la RMSD dei dati del vettore  $v$  rispetto a  $m$ —ovvero l'output è:  $\sqrt{\sum_{i=0}^{n-1} (v[i] - m)^2}$ .

Si calcoli la complessità della funzione in funzione delle dimensioni di  $v$ .

UNIVERSITÀ DI UDINE  
 CDL IN MATEMATICA  
 PROVA SCRITTA DI INFORMATICA 1  
 24 Giugno 2004

1. Si descriva il ciclo di funzionamento di una CPU e le principali tipologie di operazioni ASSEMBLER.
2. Di quanti Bytes di memoria necessiterebbe la digitalizzazione di un film muto della durata di un'ora usando colori RGB, di definizione  $640 \times 480$  pixels a 25 fotogrammi al secondo, se non si disponesse di opportuni algoritmi di compressione? Se lo stesso film fosse convertito in formato Divx e memorizzato in un CDROM, quale sarà il rapporto di compressione?<sup>1</sup>
3. Un cagnolino ha scritto correttamente il suo nome di 4 lettere su un file ASCII. Il suo padrone l'ha stampato interpretandolo erroneamente come un numero a virgola mobile su 4 Bytes. Il numero è:

$$1.10010010100110001001111 \cdot 2^{25}$$

dove a mantissa è espressa in binario (se preferite, partite da 1.5726412534713745 in decimale). Come si chiama il cagnolino?<sup>2</sup>

4. Qual è l'output del seguente programma, dati i valori di  $x$  e  $y$ ?

```
int t;
main(){
    void p(int, int);
    int i,x,y;
    scanf("%d",&x);
    scanf("%d",&y);
    t = 0;
    for(i = 1; i <= x; i++)
        p(x,y);
    printf("%d,%d,%d,%d",x,y,t,&t);
}

void p(int a, int b){
    int x, y;
    y = t-a;
    for(x = 1; x <= b; x++)
        t++;
}
```

5. Un quadrato magico di dimensione  $N$  è una matrice di dimensione  $N \times N$  tale che

- Nelle celle della matrice ci sono tutti e soli i numeri  $1, 2, 3, \dots, N^2$ .
- La somma dei numeri di ogni riga e di ogni colonna è costante ed è pari alla somma dei numeri in ciascuna delle due diagonali.

2	7	6
9	5	1
4	3	8

1	2	15	16
12	14	3	5
13	7	10	4
8	11	6	9

1	2	13	24	25
3	23	17	6	16
20	21	11	8	5
22	4	14	18	7
19	15	10	9	12

Figure 1: Esempi di quadrati magici con  $N = 3, 4, 5$

Si scriva una funzione booleana in C che accetta in input una matrice di interi di dimensione  $N \times N$  e restituisce `true` se e solo se la matrice rappresenta un quadrato magico.

Si studi la complessità della funzione descritta.

<sup>1</sup>Ovvero il rapporto tra la dimensione del file originale e la dimensione del file Divx.

<sup>2</sup>si ricorda che il codice ASCII del carattere 'A' è 65.

UNIVERSITÀ DI UDINE  
CDL IN MATEMATICA  
PROVA SCRITTA DI INFORMATICA 1  
7 Luglio 2005

1. Si illustrino e descrivano in breve i principali supporti per la memoria secondaria.
2. Si illustri come un frammento di diagramma di flusso rappresentante un generico costruito `if / then / else` possa essere simulato mediante un frammento di diagramma di flusso fortemente strutturato.
3. Si rappresenti, con la notazione in virgola mobile IEEE su 32 bit, il numero 204,53515625
4. Si spieghi qual è l'output del seguente programma C in funzione del valore `n` letto dall'input. Se ne calcoli dunque la complessità computazionale in funzione del valore di `n`:

```
main(){
    void a (int,int[]);
    int n;
    int v [1] = {0};
    scanf ("%d",&n);
    a(n,v);
    printf ("%d, %d", n , v[0]);
}
void a(int x, int v []){
    void b(int,int,int []);
    b(x, x, v);
    x = x - 1;
}
void b(int y, int z, int v []){
    void a(int,int[]);
    if (y > 0) {
        v[0] = v[0] + 1;
        b(y-1, z, v);
    }
    else if (z > 0)
        a(z-1,v);
}
```

5. Si considerino dei vettori di caratteri di  $N$  elementi, che chiameremo per semplicità *stringhe*. Tali vettori sono usati per inserire dei testi. Si assume che un testo sia finito quando, scandendo il vettore dalla cella 0 in su, si incontra il simbolo '\$'. Immaginiamo dunque il vettore `v` tale che `v[0]= 'a'`, `v[1]= 'b'`, `v[2]= '$'`, ... come la sequenza di caratteri `ab$....` Si scriva una funzione `C` che accetta come parametri due stringhe `u` e `v` siffatte e restituisce:

- Se la stringa identificata da `u` (eccetto il '\$') si ritrova come sottostringa di `v` a partire dalla posizione `i` (e non vi è un nessun indice minore per cui ciò accade), restituisce `i`.
- `-1` altrimenti.

Ad esempio:

- per `u = esame$....`, `v = oggi_supero_l'esame...` restituisce 14,
- per `u = esame$....`, `v = esame_esame_esame$` restituisce 0,
- per `u = esame$....`, `v = oggi_ho_perso_il_treno$` restituisce -1.

UNIVERSITÀ DI UDINE  
CDL IN MATEMATICA  
PROVA SCRITTA DI INFORMATICA 1  
18 Luglio 2005

1. Si discuta della rappresentazione di immagini in bianco e nero, a livelli di grigio e a colori nel calcolatore, con particolare riferimento alla quantità di memoria necessaria per immagini a  $1024 \times 768$  pixels.
2. Si descriva con diagrammi di flusso l'algoritmo per la conversione di un numero rappresentato in complemento a due a 8 bits nel numero decimale equivalente. Si supponga che l'input sia della forma  $n_7n_6 \dots n_1n_0$ .
3. Si descriva il funzionamento del seguente programma  $C$ :

```
main(){
    int proca(int,int []);
    int n, v[1] = {0};
    scanf("%d",&n);
    printf(" %d:", n); printf(" %d:", v[0]);
    printf(" %d:", proca(n,v));
    printf(" %d:", n);printf(" %d:", v[0]);
}

int proca(int n,int w[]){
    if (n > w[0]){
        w[0]++;
        return n * proca(n,w);
    }
    else {
        n++;
        return 1;
    }
}
```

4. Un insieme  $S \subseteq \mathbb{N}$  è detto *sum-free* se vale

$$\forall x \forall y ((x \in S \wedge y \in S) \rightarrow x + y \notin S)$$

Una partizione  $P = \{B_1, \dots, B_k\}$  di  $\{1, \dots, n\}$  è *buona* se ognuno dei blocchi  $B_i$  della partizione è sum-free. Una partizione siffatta si può rappresentare mediante un vettore  $v$  di almeno  $n + 1$  elementi tale che  $v[i] = j$  se e solo se il numero  $i$  appartiene al blocco  $B_j$  (per semplicità  $v[0]$  non viene usato).

Si scriva una funzione booleana  $C$  che accetta in input un vettore di interi di lunghezza maggiore a  $n$ , un numero  $n$  ed un numero  $k$  e verifica se il vettore rappresenta una partizione buona dell'insieme  $\{1, \dots, n\}$  in  $k$  blocchi (si ricorda che in una partizione non ci possono essere blocchi vuoti).

Ad esempio:  $v = [-, 1, 2, 2, 1]$  è un vettore che rappresenta la partizione  $\{\{1, 4\}, \{2, 3\}\}$  che è buona ( $n = 4$  e  $k = 2$ )—con  $-$  indico che il valore della cella 0 non interessa. Invece  $v = [-, 1, 2, 2, 1, 1]$  rappresenta la partizione  $\{\{1, 4, 5\}, \{2, 3\}\}$  che non è una buona partizione di  $\{1, 2, 3, 4, 5\}$ .

Come curiosità, non si sa se esista una buona partizione per  $n = 161$  e  $k = 5$ .



UNIVERSITÀ DI UDINE  
CDL IN MATEMATICA  
PROVA SCRITTA DI INFORMATICA 1  
21 Giugno 2006

1. Si mostri che esiste una funzione  $f : \mathbb{N} \rightarrow \mathbb{N}$  che non può essere calcolata da nessun calcolatore elettronico.
2. Si dispone di una registrazione di un concerto di 75 minuti nel formato audio stereo “wav” con la frequenza di campionamento usata nei CD audio con memorizzazione a 16 bits per canale. Qual è il rapporto di compressione che dobbiamo raggiungere con il formato “mp3” per memorizzarlo in una chiavetta USB da 128MB?
3. Siamo in un contesto di aritmetica a complemento a due su 8 bits. La somma di  $78 + B$  fornisce il risultato *erroneo* -63. Quanto vale B ?
4. Qual è l’output del seguente programma, dato il valore di x?

```
int t;
main(){
    void p(int);
    int i,x ;
    scanf("%d",&x);
    for(i = 1; i <= x; i++)
        p(i);
    printf("%d ", t);
}
void p(int a){
    int q(int);
    if (a==1)
        t = 0;
    else t = t + q(a);
}
int q(int a){
    int t;
    t = 1;
    return t + a;
}
```

5. L’ultimo teorema di Fermat asserisce che per ogni  $n \in \mathbb{N}$ ,  $n > 2$  l’equazione  $x^n + y^n = z^n$  non ammette soluzioni con  $x, y, z \in \mathbb{N} \setminus \{0\}$ .

Si scriva una funzione booleana in C che accetta in input  $m$  e  $w$  e verifica esaustivamente la congettura per tutti gli  $x, y, z \leq w$  e per tutti gli  $n \in \mathbb{N}$   $2 \leq n \leq m$ . Si studi inoltre la complessità della funzione rispetto ai valori di  $m$  e  $w$ .

UNIVERSITÀ DI UDINE  
 CDL IN MATEMATICA  
 PROVA SCRITTA DI INFORMATICA 1  
 6 Luglio 2006

1. Si illustri in breve la differenza tra compilazione ed interpretazione e si spieghino i principali vantaggi e svantaggi dei due metodi.
2. Si descriva mediante diagramma di flusso fortemente strutturato un “algoritmo” per la ricerca di soluzioni intere e positive di equazioni di terzo grado a coefficienti interi.
3. Una email contenente apparentemente una sequenza di numeri floating point rappresentati su 32 bits è stata intercettata dalla polizia postale. Grazie a una soffiata, è stato congetturato l’impiego di un codice (poco) segreto per cui, letto il primo numero come sequenza di caratteri ASCII, esso rappresenta la stringa ‘BOMB’(ciò fa sospettare che il resto della email contenga informazioni sospette). Qual è questo numero (espresso in decimale e in notazione scientifica) ?
4. Si scriva l’output del seguente programma in funzione di  $x$  e se ne determini la complessità.

```
int t;
main(){
    int p(int,int,int); int x ; t = 0;
    scanf("%d",&x);
    printf("%d\n", p(x,x,x) );printf("%d\n", x );printf("%d\n", t );
}
int p(int x, int y, int z){
    t++;
    if (x > 0){
        x = x -1; return p(x, y , z);
    }
    else if(y > 0){
        return p(y-1,y-1,z);
    }
    else return x + y + z;
}
```

5. Una cartina geografica si può rappresentare mediante una matrice  $M$  di interi in cui se tra la città  $i$  e la città adiacente  $j$  ci sono  $T$  Km, allora  $M[i, j] = T$ . Se due città  $i$  e  $j$  non hanno una strada che le collega direttamente, invece, si avrà  $M[i, j] = -1$ .

Si scriva una procedura  $C$  che prende come input una matrice del tipo appena descritto di dimensioni  $N \times N$ , una città di partenza e arrivo  $p$  e che calcola quale sia il tragitto di distanza minima per visitare altre due città (che saranno stampate, assieme al numero di Km del tragitto) e ritornare nella città  $p$  senza passare mai per la stessa strada due volte.

Nel seguente esempio, partendo da udine, la procedura stamperebbe:

#udine, #paperopoli, #ocopoli, #udine, distanza: 14 Km

ove per #citta' si intende l'intero (in questo caso da 0 a 3) associato alla città per indirizzare la matrice.

	udine	topolinia	paperopoli	ocopoli
udine	0	4	5	4
topolinia	4	0	6	-1
paperopoli	5	6	0	5
ocopoli	4	-1	5	0

UNIVERSITÀ DI UDINE  
CDL IN MATEMATICA  
PROVA SCRITTA DI INFORMATICA 1  
20 Luglio 2006

1. Si presentino, anche utilizzando opportuni esempi, le metodologie di passaggio parametri in C.
2. Si descriva, mediante diagramma di flusso *fortemente strutturato* un algoritmo per il calcolo della funzione  $\lfloor \log_2 n \rfloor : \mathbb{N} \rightarrow \mathbb{N}$ .
3. Si esegua la seguente somma in aritmetica a complemento a due su 16 bits:  $-1 + 16707$ . Si legga dunque la sequenza di bits del risultato come sequenza di due caratteri ASCII.
4. Si studi l'output in funzione di  $n$  del seguente programma C:

```
main(){
    int fun(int, int, int);
    int n;
    scanf("%d",&n);
    printf("%d", fun(n,n,n));
}

int fun(int x, int y, int z){
    if (x > 0)
        return 1 + fun(x-1,y,z);
    else if (y > 0)
        return 1 + fun(y-1,y-1,z);
    else if (z > 0)
        return 1 + fun(0,0,z-1);
    else return 0;
}
```

5. Il gioco Kisuda, versione estiva di un più noto passatempo di massa, consiste in una matrice quadrata  $N \times N$  che può contenere numeri naturali da 1 a  $N$ . Alcuni numeri sono fissati. L'obiettivo è quello di assegnare i valori alle celle non ancora fissate in modo tale che in ogni *riga*, in ogni *colonna*, e nelle due *diagonali* vi siano tutti i numeri da 1 a  $N$ .

Si scriva una funzione booleana in C che accetta in input una matrice  $N \times N$  e *verifica* se i valori in essa contenuti costituiscono una soluzione ammissibile per il gioco Kisuda. Si studi la complessità della funzione definita.

UNIVERSITÀ DI UDINE  
CDL IN MATEMATICA  
PROVA SCRITTA DI INFORMATICA 1  
5 Settembre 2006

1. Si illustri il ciclo di funzionamento della CPU
2. Si descrivano alcuni supporti di memoria secondaria facendo, se possibile, riferimento, a grandi linee, alla capacità di ognuno di essi.
3. Si rappresenti mediante la rappresentazione a virgola mobile a 32 bit, il numero 1023,49. A quanto ammonta l'errore dovuto alla rappresentazione?
4. Si studi la complessità in funzione di  $n$  della funzione  $C_p$ :

```
int p(int n){
    int q(int, int);
    return q(1,n);
}
int q(int a, int b){
    if (a > b)
        return 1;
    else
        return 1 + q(a + 1 + (b+a)/3,b);
}
```

5. Si scriva una procedura  $C$  che accetta *come parametro* un numero naturale  $n$  “non troppo piccolo né troppo grande” e stampa sul video la seguente figura (di altezza  $2n+1$  e larghezza  $2n+1$ —nell'esempio  $n = 4$ ):

```
    *
   *o*
  *ooo*
 *oooo*
*****
   *
   *
   *
   *
}
```

UNIVERSITÀ DI UDINE  
CDL IN MATEMATICA  
PROVA SCRITTA DI INFORMATICA 1  
20 Settembre 2006

1. Si dimostri che esistono funzioni da  $\mathbb{N}$  in  $\mathbb{N}$  non calcolabili.
2. Si fornisca un limite superiore della codifica della sola parte video, di un film (a colori) con la definizione di  $800 \times 600$  pixels, a 25 fotogrammi al secondo, della durata di 100 minuti. Se viene memorizzato su un supporto CD, di quante volte è stato compresso?
3. Si sommino, dopo averli trasformati in virgola mobile su 32 bits, 0.3 e 0.6. A quanto ammonta l'errore dovuto alla rappresentazione?
4. Si studi la complessità in funzione di  $n$  della funzione C p:

```
int p(int n){
    int q(int, int);
    return q(1,n);
}
int q(int a, int b){
    if (a > b)
        return 1;
    else
        return 1 + q(a + 1 + (b-a)/3,b);
}
```

5. Si scriva una funzione C che accetta *come parametro* un numero naturale  $n$  e stabilisce se  $n$  è un prodotto di due numeri primi. In tal caso restituisce come output (non stampa) la somma di questi due numeri, altrimenti restituisce 0. Si studi la sua complessità. Se necessario, si definiscano delle procedure/funzioni ausiliarie.