

Università della Calabria
Corsi di Laurea in Chimica, Fisica e Scienze dei Materiali

Esame di Informatica

(Variante A)

Non è consentito l'uso di alcun tipo di documentazione e strumentazione elettronica.
Si utilizzi il retro di queste pagine per svolgere i calcoli e implementare i programmi richiesti.
Tempo a disposizione [hh:mm]: **1:30**

Cognome: _____ Nome: _____ Matricola _____

Corso di Studi: Chimica Fisica Scienze dei Materiali Data [gg/mm/aaaa]: _____

1. [2 punti] Si converta in notazione binaria il numero naturale decimale ottenuto sommando al numero 21 l'ultima cifra del proprio numero di matricola. Qualora l'ultima cifra fosse 0, sommare 5. Per esempio, se la matricola è 826454, convertire in binario il numero 25. Se, invece, la matricola fosse 826570, convertire il numero 26.

Soluzione: La soluzione va scritta in questo modo: $23_{10} = 10111_2$

2. [1 punto] I numeri interi binari con n bit variano nel seguente intervallo:

- $[-2^{n-1} - 1, 2^{n-1}]$
 $[-2^{n-1}, 2^{n-1} - 1]$
 $[0, 2^{n-1} - 1]$
 $[0, 2^n - 1]$

3. [2 punti] Il numero intero binario 1110 in complemento a 2 con quattro bit corrisponde a:

- 10
 -6
 -4
 -2

4. [1 punto] L'Unità di controllo di un calcolatore di von Neumann esegue ripetutamente i seguenti step fondamentali:

- Fetch ed esecuzione delle istruzioni del programma;
 Fetch, decodifica ed esecuzione delle istruzioni del programma;
 Fetch, decodifica, esecuzione e store delle istruzioni del programma;
 Pre-fetch, fetch, decodifica, esecuzione e store delle istruzioni del programma;

5. [1 punto] Quale delle seguenti affermazioni è vera a proposito dei differenti tipi di memoria nei calcolatori di von Neumann?

- La RAM contiene istruzioni e dati, i registri dell'ALU operandi di operazioni aritmetico-logiche;**
 La RAM contiene istruzioni, i registri dell'unità di controllo i dati;
 La RAM contiene i dati, i registri dell'unità di controllo le istruzioni;
 La RAM contiene gli operandi, i registri dell'ALU le istruzioni aritmetico-logiche.

Esame di Informatica

(Variante A)

6. [3 punti] Si consideri la seguente successione di numeri naturali:

$$a_1 = 1$$

$$a_2 = 2$$

$$a_n = a_{n-2} + a_{n-1} + 1$$

Si scriva una funzione che riceva in input un intero $n \geq 1$ e ritorni l' n -esimo termine della successione. La funzione ritorni 0 se $n < 1$. L'algoritmo non utilizzi oggetti array per la risoluzione del problema.

Soluzione:

```
int termineSuccessione(const int n)
{
    if (n < 1) return 0;
    if (n == 1) return 1;
    if (n == 2) return 2;

    int c = 1;
    int b = 2;
    int a;
    for (int i=3; i<=n; i++)
    {
        a = c + b + 1;
        c = b;
        b = a;
    }

    return a;
}
```

7. [3 punti] Scrivere una funzione C++ che riceva in input un riferimento costante a `double` e ritorni 0.0 se e solo se esso sia compreso tra -10^{-3} e 10^{-3} . In caso contrario, ritorni il valore del parametro.

Soluzione:

```
int annullamentoCondizionato(const double& a)
{
    if (-1.0e-3 <= a && a <= 1.0e-3)
        return 0;

    return a;
}
```

Esame di Informatica

(Variante A)

8. [4 punti] Scrivere una funzione che riceva in input un array dinamico `v` di valori `double`, deallochi il buffer associato (cioè la sequenza di `double` associata al puntatore `v`) e imposti `v` a `NULL`, facendo attenzione che anche l'argomento (cioè l'oggetto utilizzato all'invocazione della funzione per definire `v`) punti a `NULL` al termine dell'esecuzione della funzione.

Soluzione:

```
void deleteArray(double*& v)
{
    delete [] v;
    v = NULL;
}
```

9. [5 punti] Scrivere una funzione che riceva in input due array di valori interi `v1` e `v2` di lunghezza `l1` ed `l2` e costruisca un terzo array `v3` contenente i soli valori dispari degli array `v1` e `v2` prelevando i valori prima dall'array `v1` e, una volta esauriti, dall'array `v2`. La funzione deve ritornare al chiamante sia l'array `v3` che sua la dimensione.

Soluzione:

```
void dispari(const int * const v1, const int l1, const int * const v2, const int
            l2, int*& v3, int& l3)
{
    int l1d = 0;
    int l2d = 0;

    for (int i=0; i<l1; i++)
        if (v1[i]%2)
            l1d++;
    for (int i=0; i<l2; i++)
        if (v2[i]%2)
            l2d++;

    l3 = l1d+l2d;
    if (l3 == 0)
        return;

    v3 = new (nothrow) int [l3];

    int j = 0;
    for (int i=0; i<l1; i++)
        if (v1[i]%2)
            v3[j++] = v1[i];
    for (int i=0; i<l2; i++)
        if (v2[i]%2)
            v3[j++] = v2[i];
}
```

Esame di Informatica

(Variante A)

10. [7 punti] Scrivere una funzione che riceva in input un array v di valori binari di lunghezza $L \geq 3$ e verifichi se esiste una sequenza di lunghezza $l \leq L$ tale che, interpretata come numero naturale decimale, ricada nell'intervallo $[i1, i2]$, con $0 \leq i1 \leq i2 \leq 7$.

Ad esempio, se $v = 1,1,1,0,1,1,0,1$, $L = 8$, $l = 3$, $i1 = 2$ e $i2 = 4$, allora la sequenza $0,1,1$ soddisfa la condizione poichè equivale al numero decimale 3. Nessuna altra sequenza soddisfa la condizione.

Soluzione:

```
int convert(int* v, int L, int i, int offset)
{
    if (i+offset >= L)
        return -1;

    int decimal = 0.0;

    for (int e=offset; e>=0; e--)
        decimal += v[i+offset-e] * pow(2,e);

    return decimal;
}

bool controllaArray(const int * const v, const int L, const int l, const int i1,
                  const int i2)
{
    for (int i=0; i<=L-l; i++)
    {
        int decimal = convert(v, L, i, l-1);
        if (decimal >= i1 && decimal <= i2)
            return true;
    }

    return false;
}
```

11. [7 punti] Scrivere una funzione che riceva in input una matrice dinamica 4x4 di elementi interi e verifichi se esiste una sottomatrice nulla 2x2.

Soluzione:

```
bool checkNull2(float** M, int r, int c)
{
    for (int i=r; i<r+2; i++)
        for (int j=c; j<c+2; j++)
            if (M[i][j] != 0)
                return false;

    return true;
}

bool findNull2(float** M)
{
    const int dim = 4;
    for (int r=0; r<=dim-2; r++)
        for (int c=0; c<=dim-2; c++)
            if (checkNull2(M, r, c))
                return true;

    return false;
}
```