

Sperimentazioni di Fisica I (mod. A)

Prova di Esonero di Informatica del 06.02.2020

Tempo consentito: 1 ora e 30 minuti

Cognome:	
Nome:	Matricola:

Non è consentito l'utilizzo di calcolatrici, appunti e quaderni. Verrà valutato solo quanto riportato in questi fogli, eventuali "fogli di brutta" non verranno valutati. Negli esercizi a scelta multipla, l'indicazione del risultato senza lo svolgimento scritto dell'esercizio verrà valutato 0 punti.

Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Totale

1. [4 punti] Dati i numeri in base 10 $A = -24.375$ e $B = +12.2$, rappresentarli in base 2, complemento a due (C_2), in virgola fissa con notazione Q6.2 ed effettuare le operazioni $A + B$ e $A - B$. Indicare la risposta corretta:

Risposta 1, $A + B =$

- 111100110 111010111 111001111 111011111 nessuna delle precedenti

Risposta 2, $A - B =$

- 101101011 101101111 101101100 101101101 nessuna delle precedenti

CONVERSIONI:

$24 : 2 = 12$ 0
 $12 : 2 = 6$ 0
 $6 : 2 = 3$ 0
 $3 : 2 = 1$ 1
 $1 : 2 = 0$ 1

$0,375 \times 2 = 0,75$
 $0,75 \times 2 = 1,5$
 $0,5 \times 2 = 1,0$

12
 ↓
 1100
 vedi conversione
 24

$0,2 \times 2 = 0,4$
 $0,4 \times 2 = 0,8$
 $0,8 \times 2 = 1,6$

$+|A| = \underset{S}{0} \underset{P.I.}{011000} \underset{P.D.}{01}$

$B = 0 \ 001100 \ 00$

$A = 1 \ 100111 \ 11$

$-B = 1 \ 110100 \ 00$

$A+B$
 $1 \ 100111 \ 11 +$
 $0 \ 001100 \ 00 =$

 $1 \ 110011 \ 11$

$A-B$ (11) → no overflow
 $1 \ 100111 \ 11 +$
 $1 \ 110100 \ 00 =$

 $1 \ 011011 \ 11$

2. [2.5 punti] Dato il numero decimale $x = -125_{10}$, rappresentarlo in base binaria nelle seguenti notazioni, utilizzando (complessivamente) 9 bit:

125 : 2 = 62 1
 62 : 2 = 31 0
 31 : 2 = 15 1
 15 : 2 = 7 1
 7 : 2 = 3 1
 3 : 2 = 1 1
 1 : 2 = 0 1

130 : 2 = 65 0
 65 : 2 = 32 1
 32 : 2 = 16 0
 16 : 2 = 8 0
 8 : 2 = 4 0
 4 : 2 = 2 0
 2 : 2 = 1 0
 1 : 2 = 0 1

- Modulo e Segno M_S in base due 101111101
s → aggiunge uno zero 9 bit
- Complemento a Uno C_1 in base due $+|x| = 001111101$ $\xrightarrow{C_1} 110000010$
- Complemento a Due C_2 in base due $C_2 = C_1 + 1$ 110000011
- Eccesso-q $q = 2^{n-1} - 1 = 2^8 - 1 = 255$
 $n = 255 - 125 = 130$ $\xrightarrow{\text{converta}}$ 10000010 $\xrightarrow{\text{3bit}}$ 010000010

Risposta 1, $M_S =$

- 110011001 101111101 110000111 110001111 nessuna delle precedenti

Risposta 2, $C_1 =$

- 101110000 101100110 110000010 101111000 nessuna delle precedenti

Risposta 3, $C_2 =$

- 110000011 101111001 101110001 101100111 nessuna delle precedenti

Risposta 4, Eccesso-q =

- 001110000 001100110 010000010 001111000 nessuna delle precedenti

3. [4 punti] Interpretare la sequenza di 9 caratteri binari 110110110 come la rappresentazione di un numero in base 2, complemento a due (C_2), in virgola fissa con notazione Q5.3. Interpretare successivamente la medesima sequenza come la rappresentazione in virgola mobile di un numero in base 2, utilizzando 1 per il segno, 4 per l'esponente e 4 per la mantissa. Scrivere i corrispondenti numeri in base 10.

Risposta 1, Q5.3:

- 9.25 -5.375 -7.25 -12.875 nessuna delle precedenti

Risposta 2, virgola mobile:

- 84 -22 -44 -6.25 nessuna delle precedenti

$1 \ 10110 \ 110$
 s → negativo
 faccio C_2

$0 \ 01001 \ 010$
 P.I. P.D.

$2^3 + 2^0 + 2^{-2} =$

$8 + 1 + 0,25 = 9,25$
 ↑
 modulo

-9,25

$1 \ 1011 \ 0110$
 s exp mantissa
 L → negativo

$exp \rightarrow (1011)_2 = (2^3 + 2^1 + 2^0)_{10} = 11_{10}$

$q = 2^{n-1} - 1 = 7$

$exp = 11 - 7 = 4$

$1,0110 \cdot 10^4$

↑
 bit nascosto

10110

$2^4 + 2^2 + 2^1 = 16 + 4 + 2 = 22$

-22

4. [2 punti] Data la seguente tabella di verità, ricavarne la corrispondente funzione booleana utilizzando il teorema fondamentale dell'algebra booleana e semplificarla usando le regole dell'algebra booleana, indicando a lato la proprietà utilizzata, ove possibile.

a	b	c	f(a,b,c)
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Risposta:

- $\bar{a}\bar{b} + \bar{c}$
- $\bar{a}\bar{b} + c$
- $ab + c$
- $ab + \bar{c}$
- nessuna delle precedenti

$$\begin{aligned} & \bar{a}\bar{b}\bar{c} + \bar{a}\bar{b}c + \bar{a}b\bar{c} + \bar{a}bc + abc \\ & \underline{\bar{a}\bar{b}\bar{c}} + \underline{\bar{a}\bar{b}c} + \underline{\bar{a}b\bar{c}} + \underline{\bar{a}bc} + \underline{abc} + \underline{\bar{a}\bar{b}c} \\ & \bar{a}\bar{b}(\bar{c}+c) + \bar{a}c(\bar{b}+b) + ac(\bar{b}+b) \\ & \bar{a}\bar{b} + \bar{a}c + ac \\ & \bar{a}\bar{b} + c(\bar{a}+a) \\ & \bar{a}\bar{b} + c \end{aligned}$$

↓ IDEMPOTENZA
 ← ASSOCIATIVA
 ← COMPLEMENTO
 $(a+\bar{a})=1$
 ELEMENTO NEUTRO
 $a*1=a$
 ← ASSOCIATIVA
 ← COMPLEMENTO ED ELEMENTO NEUTRO

5. [2.5 punti] Semplificare la seguente espressione, utilizzando le regole dell'algebra booleana:

$$\overline{abc + \bar{a}b\bar{c} + bcd + abd + \bar{a}cd + \bar{a}bd + (\bar{a}+c)d + b\bar{c}(a+d) + bcd + ab\bar{c} + a\bar{c}d}$$

Risposta:

- $\bar{a}\bar{b}$
- $\bar{a} * \bar{c}$
- $\bar{b}\bar{c}$
- $\bar{b} * \bar{d}$
- nessuna delle precedenti

$$\overline{\bar{a}bc + \bar{a}b\bar{c} + \underline{bcd} + \underline{abd} + \bar{a}cd + \bar{a}bd + \bar{a}\bar{c}d + \underline{\bar{a}b\bar{c}d} + \underline{bcd} + ab\bar{c} + a\bar{c}d}$$

$$\bar{a}bc + \bar{a}b\bar{c}(1+\bar{d}) + bc(d+\bar{d}) + ad(\bar{b}+b) + \bar{a}cd + \bar{a}\bar{c}d + ab\bar{c} + a\bar{c}d$$

$$\underline{\bar{a}bc} + \underline{\bar{a}b\bar{c}} + \underline{bc} + \underline{ad} + \underline{\bar{a}cd} + \underline{\bar{a}\bar{c}d} + \underline{ab\bar{c}} + \underline{a\bar{c}d}$$

$$bc(\bar{a}+1) + b\bar{c}(\bar{a}+a) + ad(1+\bar{c}) + \bar{a}d(c+\bar{c})$$

$$bc + b\bar{c} + ad + \bar{a}d = b(c+\bar{c}) + d(a+\bar{a}) = b+d = \bar{b} * \bar{d}$$

* si utilizza in caso di vettori e strutture di grandi dimensioni.
 ** si utilizza nel caso degli array (solitamente puntatore al primo elemento).

A.A. 2019/20

Sperimentazioni di Fisica I (mod. A)

06.02.2020

6. [2 punti] Descrivere brevemente le diverse modalità di passaggio degli argomenti in ingresso (input) ad una funzione.

Gli argomenti a una funzione possono essere passati in 3 modi: by-value, by-reference e by pointer.

Pass by value: quando chiamo la funzione viene creata una copia della variabile e le viene assegnato lo stesso valore. La funzione lavora sulla copia, non modifica la variabile.

Pass by reference: nella funzione utilizzo la stessa variabile della chiamata soltanto che avrà un nome diverso alias. Lavorando sull'alias modifico il contenuto della variabile originale. *

Pass by pointer: quando chiamo la funzione passo l'indirizzo di memoria che viene assegnato ad un puntatore nella funzione. Lavorando sugli indirizzi di memoria, modifico le variabili della...

Si utilizza per singole variabili, vettori e strutture di piccole dimensioni.

7. [4 punti] Simmetria rispetto alla bisettrice del secondo e del quarto quadrante: $y = -x$. Le coordinate di un punto $P'(x', y')$ simmetrico rispetto al punto $P(x, y)$ si trovano utilizzando la seguente trasformazione:

$$\begin{cases} x' = -y \\ y' = -x \end{cases}$$

Scrivere i prototipi e le definizioni di due funzioni che effettuano le seguenti operazioni: la prima calcolerà il simmetrico di un punto P, rispetto all'asse $y = -x$, la seconda calcolerà la distanza del punto dalla bisettrice del secondo e quarto quadrante $d = \frac{|(y_P + x_P)|}{\sqrt{2}}$. Entrambe le funzioni accetteranno in input una struttura di nome Punto, da definire per la gestione delle due coordinate x, y di un punto nel piano cartesiano (utilizzare all'interno della struttura delle variabili in virgola mobile). Ricordare di indicare gli include files necessari e di abilitare, eventualmente, il namespace.

```
#include <iostream>
#include <cmath>  -> E' NECESSARIA
using namespace std;

struct Punto {    // STRUTTURA
    double x;
    double y;
}

Punto simm (Punto); // PROTOTIPO 1
double dist (Punto); // PROTOTIPO 2

Punto simm (Punto P) // DEFINIZIONE 1
{
    Punto S;
    S.x = -P.y;
    S.y = -P.x;
    return S;
}

double dist (Punto P) // DEFINIZIONE 2
{
    double d;
    d = abs (P.y + P.x) / sqrt (2);
    return d;
}
```

8. [2 punti] Scrivere i comandi necessari nel sistema operativo Linux per effettuare le seguenti operazioni:

- Cancellare la directory (Piena) Video. `rm -r Video`
- Creare la directory Foto, `mkdir Foto`
- Entrare nella sottocartella della home Download, `cd -` per andare nella home `cd Download`
- Copiare il file Europei2016.txt nel file Europei2020.txt, `cp Europei2016.txt Europei2020.txt`
- Visualizzare le ultime 20 righe del file funzione.cxx, `tail -n 20 funzione.cxx`
- Visualizzare il nome della directory in cui si sta lavorando, `pwd`
- Ordinare, in ordine inverso, il contenuto del file ElencoStudenti.txt, `sort -r ElencoStudenti.txt`
- Rinominare il file Pressioni.txt come Volumi.txt. `mv Pressioni.txt Volumi.txt`

9. [4 punti] Scrivere nel linguaggio di programmazione C++ un programma che chieda all'utente di inserire i valori misurati dell'allungamento di una molla e ne calcoli il valore medio e la deviazione standard. Inoltre il programma fornirà il numero di dati compresi nell'intervallo $[-3\sigma, 3\sigma]$. Ricordare di indicare gli include files necessari e di abilitare, eventualmente, il namespace. Si ricordano le formule:

$$\bar{x} = \frac{1}{N} \sum_{i=0}^{N-1} x_i, \quad \sigma_x = \sqrt{\frac{\sum_{i=0}^{N-1} (x_i - \bar{x})^2}{N-1}}$$

Esempio di interfaccia del programma:

Inserire degli allungamenti: 1.01 1.02 1.01 1.00 0.99 1.01 1.20 0.99 <Ctrl D>

Sono stati inseriti 9 valori

Il valore medio del periodo e': 1.0278 cm

Il la deviazione standard e': 0.0655 cm

Il valori all'interno dall'intervallo sono: 9

```
#include <iostream>
#include <vector>
#include <cmath>
using namespace std;
int main()
{
    int cnt=0;
    double val, media=0, dev=0, dx, sx;
    vector<double> v
    cout << "inserire i valori" << endl;
    while (cin >> val)
    {
        v.push_back(val);
        media += val;
    }
    cout << "sono stati inseriti" << v.size();
    cout << "valori" << endl;
    cout << "il valore medio e'" << media;
    cout << "cm" << endl;
```

```
for (auto e: v)
{
    dev += pow((e - media), 2);
}
dev = dev / (v.size() - 1);
dx = media + 3 * dev;
sx = media - 3 * dev;
for (auto c: v)
{
    if (c < dx || c > sx) cnt++;
}
cout << "i valori nell'intervallo sono:";
cout << cnt << endl;
return 0;
```