

---

**ESAME DI FONDAMENTI DI INFORMATICA I**  
**ESAME DI ELEMENTI DI INFORMATICA**  
**17/1/1997**

**PROVA SCRITTA**

1. Mostrare come viene svolto il calcolo:

$$9.8 - 2.6 + (10 * 2^2)$$

qualora l'elaboratore adotti per i numeri interi una rappresentazione in complemento a due su 8 bit e per i numeri reali una rappresentazione in virgola mobile con un byte per la mantissa normalizzata e un byte per l'esponente in complemento a due.

Mostrare i passaggi intermedi eseguiti dall'elaboratore con particolare riferimento alle operazioni e alla rappresentazione interna in binario, nonché i relativi eventuali errori compiuti. Scrivere il risultato finale in codifica decimale.

2. Sia data la funzione logica F: "uno studente può laurearsi in ingegneria se ha sostenuto i 29 esami previsti e ha redatto una tesi di laurea, oppure è in possesso di un diploma universitario e ha sostenuto 14 esami previsti e ha redatto una tesi di laurea". In funzione delle variabili logiche:

- E1, con valore 1 se ha sostenuto i 29 esami previsti;
- E2, con valore 1 se ha sostenuto i 14 esami previsti per i diplomati;
- D, con valore 1 se è in possesso di un diploma universitario;
- T, con valore 1 se ha redatto una tesi di laurea;

si scriva l'espressione analitica di F e la tabella della verità.

---

**PROVA PRATICA**

Si scriva un programma C (BASIC per chi sostiene l'esame di Elementi di Informatica) che:

- a) legga a terminale una sequenza di interi positivi terminata dal valore 0 e li inserisca in un vettore di 20 componenti (si suppone che la sequenza sia di lunghezza minore o uguale a 20);
- b) chiami una subroutine che stampi a terminale i numeri interi contenuti nel vettore divisibili per 5.

**Esempio:**

Con ingresso:           15 3 2 7 8 9 10 60 0

Viene stampato:       15 10 60

---

**ESAME DI FONDAMENTI DI INFORMATICA I**  
**ESAME DI ELEMENTI DI INFORMATICA**  
**24/2/1997**

**PROVA SCRITTA**

1. Mostrare come viene svolto il calcolo:

$$10.9 - 3.5 + (10 * 2^2)$$

qualora l'elaboratore adotti per i numeri interi una rappresentazione in complemento a due su 8 bit e per i numeri reali una rappresentazione in virgola mobile con un byte per la mantissa normalizzata e un byte per l'esponente in complemento a due.

Mostrare i passaggi intermedi eseguiti dall'elaboratore con particolare riferimento alle operazioni e alla rappresentazione interna in binario, nonché i relativi eventuali errori compiuti. Scrivere il risultato finale in codifica decimale.

2. Data la funzione logica F: "È necessario prendere l'aspirina se si ha l'emicrania e male ai denti oppure l'emicrania, nausea e capogiri oppure se non si ha l'emicrania e si ha la febbre oppure se non si ha l'emicrania e si ha la febbre e la nausea" scrivere l'espressione analitica di F in funzione delle variabili logiche:

- Asp, con valore 1 se è necessario prendere l'aspirina;
- EM, con valore 1 se si ha l'emicrania;
- D, con valore 1 se si ha male ai denti;
- N, con valore 1 se si ha la nausea;
- F, con valore 1 se si ha la febbre.
- C con valore 1 se si hanno i capogiri.

Si scriva inoltre la tabella di verità

---

**PROVA PRATICA**

Si scriva un programma C (BASIC per chi sostiene l'esame di Elementi di Informatica) che:

- a) legga da terminale una sequenza di interi positivi terminata dal valore 0 e li inserisca in un vettore di 20 componenti (si suppone che la sequenza sia di lunghezza minore o uguale a 20);
- b) chiami una subroutine che stampi a terminale i numeri interi contenuti nel vettore divisibili per 3.

**Esempio:**

Con ingresso:           15 3 2 7 8 9 10 60 0

Viene stampato:       15 3 9 60

SOLUZIONE COMPITO FEBBRAIO

Rappresentazione mantissa + esponente

$$10,9_{10} = 01010,1110$$

normalizzo  $0,10101110 * 2^{100}$

Si considerano 8 bit per il modulo della mantissa in quanto si utilizza il primo bit della mantissa normalizzata, che è sempre ad 1 per definizione, per memorizzare il segno. Quindi nella memoria del calcolatore il numero verrebbe rappresentato come

$$00101110 \ 00000100$$

dove il primo 0 sostituisce il primo 1 della mantissa e indica che il numero è positivo. Negli esercizi seguenti la effettiva rappresentazione del numero in memoria verrà omessa per brevità. All'esame la rappresentazione in memoria può essere omessa.

$$3,5_{10} = 011,10000$$

normalizzo  $0,1110000 * 2^{010}$

Rappresentazione su 8 bit in complemento a 2

$$10 = 00001010$$

$$2^2 = 4 = 00000100$$

Operazioni:

$$\begin{array}{r} 10,9 - \quad 0,1010111 * 2^{100} - \\ 3,5 = \quad 0,0011100 * 2^{100} = \\ \hline 7,4 \quad 0,0111011 * 2^{100} \ (7,375) \end{array}$$

Normalizzo:

$$0,1110110 * 2^{011}$$

$$10 * \quad 00001010$$

$$2^2 = \quad 00000100$$

$$\hline 40 \quad 00000000$$

$$00000000$$

$$00001010$$

$$\hline 00101000$$

Normalizzazione per floating point:

$$0,101000 * 2^{110}$$

Operazione finale:

$$\begin{array}{r} 7,375 + \quad 0,1010000 * 2^{110} + \\ 40 = \quad 0,0001110 * 2^{110} = \\ \hline 47,375 \quad 0,1011110 * 2^{110} \ (47) \end{array}$$

Si sono verificati un errore di troncamento nella rappresentazione di 10,9 e un errore di incolonnamento nell'operazione  $7,375 + 40$  che, complessivamente, causano una perdita di 0,4 rispetto all'operazione rappresentata in base 10 il cui risultato è 47,4.

2)

$$F = (EM \text{ and } D) \text{ or } (EM \text{ and } N \text{ and } C) \text{ or } (\text{not } EM \text{ and } F) \text{ or } (\text{not } EM \text{ and } F \text{ and } N) = \\ = (EM \text{ and } D) \text{ or } (EM \text{ and } N \text{ and } C) \text{ or } (\text{not } EM \text{ and } F)$$

```

/*prende un vettore di interi e stampa i numeri interi divisibili per 3*/

#include <stdio.h>
#define MAX 20

main()
{
    int DIGIT[MAX];
    int i,j;
    void divisibili_per_3(int n[], int lim);
    i=0;

    printf("\n");
    do
    {
        printf("Inserisci il numero %d: ",i);
        scanf("%d", &DIGIT[i]);
        i=i+1;
    }
    while (DIGIT[i-1]!=0);
    /* i-2 e' la posizione dell'ultimo numero inserito */
    printf("Il vettore risultato e':\n");
    stampa_divisibili_per_3(DIGIT, i-1);
}

void stampa_divisibili_per_3(int n[], int lim)
{
    int i;
    for(i=0;i<lim;i++)
        if (n[i]/3*3==n[i])
            printf("%d ",n[i]);
}

```

---

**ESAME DI FONDAMENTI DI INFORMATICA I**  
**ESAME DI ELEMENTI DI INFORMATICA**  
**18/3/1997**

**PROVA SCRITTA**

1. Mostrare come viene svolto il calcolo:

$$5.6 - 1.5 + (10 * 2^2)$$

qualora l'elaboratore adotti per i numeri interi una rappresentazione in complemento a due su 8 bit e per i numeri reali una rappresentazione in virgola mobile con un byte per la mantissa normalizzata e un byte per l'esponente in complemento a due.

Mostrare i passaggi intermedi eseguiti dall'elaboratore con particolare riferimento alle operazioni e alla rappresentazione interna in binario, nonché i relativi eventuali errori compiuti. Scrivere il risultato finale in codifica decimale.

2. Scrivere la tabella della verità dell'espressione  $F = a \text{ exor } b \text{ or not } (a \text{ or } b \text{ and } c)$ .

---

**PROVA PRATICA**

Si scriva un programma C (BASIC per chi sostiene l'esame di Elementi di Informatica) che:

- a) legga da terminale una sequenza di nomi propri (ciascuno su una riga nuova) terminata dalla stringa 'ok' e li inserisca in un vettore NOMI di 20 componenti (si suppone che la sequenza sia di lunghezza minore o uguale a 20);
  - b) stampi il vettore NOMI
-

SOLUZIONE COMPITO MARZO

Rappresentazione mantissa + esponente

$$5,6_{10} = 101,1001100 \quad 0,6 \text{ è un numero periodico}$$

$$\text{normalizzo } 0,10110011 * 2^{011}$$

$$1,5_{10} = 001,10000$$

$$\text{normalizzo } 0,11000000 * 2^{001}$$

Rappresentazione su 8 bit in complemento a 2

$$10 = 00001010$$

$$2^2 = 4 = 00000100$$

Operazioni:

$$\begin{array}{r} 5,6 - \\ 1,5 = \\ \hline 4,1 \end{array} \quad \begin{array}{r} 0,10110011 * 2^{011} - \\ 0,00110000 * 2^{011} \\ \hline 0,10000011 * 2^{011} \end{array} \quad (4,09375) \quad 100,00011$$

$$\begin{array}{r} 10 * \\ 2^2 = \\ \hline 40 \end{array} \quad \begin{array}{r} 00001010 \\ 00000100 \\ \hline 00000000 \\ 00000000 \\ 00001010 \\ \hline 00101000 \end{array}$$

Normalizzazione per floating point:

$$0,10100000 * 2^{110}$$

$$5,6+1,5 = 0,10000011 * 2^{011} \text{ aumento l'esponente a } 110 \rightarrow 0,00010000 * 2^{110}$$

Operazione finale:

$$\begin{array}{r} 40 \\ 4,09375 \\ \hline 44,09375 \end{array} \quad \begin{array}{r} + \\ = \\ \hline \end{array} \quad \begin{array}{r} 0,10100000 * 2^{110} \\ 0,00010000 * 2^{110} \\ \hline 0,10110000 * 2^{110} \end{array} \quad (44)$$

Si sono verificati un errore di troncamento nella rappresentazione di 5,6 e un errore di incolonnamento nell'operazione  $4,09375 + 40$  che causa una perdita complessiva di 0,1 rispetto all'operazione rappresentata in base 10 il cui risultato è 44,1.

2)

$F = a \text{ exor } b \text{ or not } (a \text{ or } b \text{ and } c)$

a	b	c	a exor b not (a or b and c)	F
0	0	0	0	1
0	0	1	0	1
0	1	0	1	1
0	1	1	1	0
1	0	0	1	0
1	0	1	1	0
1	1	0	0	0
1	1	1	0	0

```
/* Legge da terminale una sequenza di nomi propri e li stampa */

#include<stdio.h>
#include<string.h>

main()
{
    char nomi[20][20];      /*vettore che conterra' la lista dei nomi*/
    int i,n;
    char temp[20];         /*vettore temporaneo per rilevare la stringa di
                           terminazione*/

    i=0;
    do
    {
        printf("Introduci un nome: ");
        scanf("%s",temp);  /*introduce una stringa in nomi*/
        strcpy(nomi[i],temp); /*copia la stringa in temp*/
        i=i+1;
    }
    while (strcmp(temp,"ok"));
    /*condizione di terminazione inserimento nomi*/

    for(n=0;n<i;n++) /*stampa nomi*/
        printf("%s\n",nomi[n]);
}
```



---

**ESAME DI FONDAMENTI DI INFORMATICA I**  
**ESAME DI ELEMENTI DI INFORMATICA**  
**16/4/1997**

**PROVA SCRITTA**

1. Mostrare come viene svolto il calcolo:

$$5.6 - 1.5 + (10 * 2^2)$$

qualora l'elaboratore adotti per i numeri interi una rappresentazione in complemento a due su 8 bit e per i numeri reali una rappresentazione in virgola mobile con un byte per la mantissa normalizzata e un byte per l'esponente in complemento a due.

Mostrare i passaggi intermedi eseguiti dall'elaboratore con particolare riferimento alle operazioni e alla rappresentazione interna in binario, nonché i relativi eventuali errori compiuti. Scrivere il risultato finale in codifica decimale.

2. Scrivere la tabella della verità dell'espressione  $F = a \text{ exor } b \text{ or not } (a \text{ or } b \text{ and } c)$ .

---

**PROVA PRATICA**

Si scriva un programma C (BASIC per chi sostiene l'esame di Elementi di Informatica) che:

- a) legga da terminale una sequenza di interi positivi terminati dal valore 0 (uno su ogni linea) e li inserisca in un vettore DIGIT di 20 componenti (si suppone che la sequenza sia di lunghezza minore o uguale a 20);
  - b) Stampi la media: a tale scopo si scriva una funzione per il calcolo della media.
-

```

/*prende un vettore di interi e ne calcola la media */

#include <stdio.h>
#define MAX 20

main()
{
    int DIGIT[MAX];
    int i,j;
    float media(int n[], int lim);

    i=0;
    printf("\n");
    do
    {
        printf("Inserisci il numero %d: ",i);
        scanf("%d", &DIGIT[i]);
        i=i+1;
    }
    while (DIGIT[i-1]!=0);
/* i-2 e' la posizione dell'ultimo numero inserito */
    printf("La media vale: %4.2f\n",media(DIGIT,i-1));
}

float media(int n[], int lim)
{
    int i;
    float media;
    media=0;
    for(i=0;i<lim;i++)
        media=media+n[i];
    return media/lim;
}

```

---

**ESAME DI FONDAMENTI DI INFORMATICA I**  
**ESAME DI ELEMENTI DI INFORMATICA**  
**14/5/1997**

**PROVA SCRITTA**

1. Mostrare come viene svolto il calcolo:

$$7.3 - 4.7 + (6 * 2^3)$$

qualora l'elaboratore adotti per i numeri interi una rappresentazione in complemento a due su 8 bit e per i numeri reali una rappresentazione in virgola mobile con un byte per la mantissa normalizzata e un byte per l'esponente in complemento a due.

Mostrare i passaggi intermedi eseguiti dall'elaboratore con particolare riferimento alle operazioni e alla rappresentazione interna in binario, nonché i relativi eventuali errori compiuti. Scrivere il risultato finale in codifica decimale.

2. Sia data la funzione logica F: "un cittadino italiano deve pagare il 20% del suo reddito in tasse se ha un reddito inferiore ai 25 milioni all'anno e ha tre familiari a carico, oppure se ha un reddito inferiore ai 30 milioni e ha 2 familiari a carico oppure se è un pensionato con altri redditi oltre alla pensione". In funzione delle variabili logiche:

- R25, con valore 1 se ha un reddito inferiore a 25 milioni;
- R25-30, con valore 1 se ha un reddito compreso tra i 25 e i 30 milioni;
- F3, con valore 1 se ha tre familiari a carico;
- F2, con valore 1 se ha due familiari a carico;
- P, con valore 1 se è pensionato;
- R, con valore 1 se non ha altri redditi oltre alla pensione.

scrivere l'espressione analitica di F e la tabella della verità.

---

**PROVA PRATICA**

Si scriva un programma C (BASIC per chi sostiene l'esame di Elementi di Informatica) che:

- a) legga da terminale una sequenza di interi positivi pari terminati dal primo valore dispari (uno su ogni linea) e li inserisca in un vettore DIGIT di 20 componenti (si suppone che la sequenza sia di lunghezza minore o uguale a 20);
  - b) Stampi a terminale il risultato della divisione per 2 di ciascun numero.
-

SOLUZIONE COMPITO MAGGIO

Rappresentazione mantissa + esponente

$7,3_{10} = 111,010011$        $0,6$  è un numero periodico  
 normalizzo  $0,11101001 * 2^{011}$

$4,7_{10} = 100,1011001$   
 normalizzo  $0,10010110 * 2^{011}$

Rappresentazione su 8 bit in complemento a 2

$6 = 00000110$

$2^3 = 8 = 00001000$

Operazioni:

$7,3 -$	$0,11101001 * 2^{011} -$
$4,7 =$	$0,10010110 * 2^{011}$
-----	-----
$2,6$	$0,01010011 * 2^{011} =$
normalizzo	$0,10100110 * 2^{010} = 2.59375$ Errore di troncamento

$6 *$	$00000110$
$2^3 =$	$00001000$
-----	-----
$48$	$00000000$
	$00000000$
	$00000000$
	$00000110$
	-----
	$0000011000$

Normalizzazione per floating point:

$0,11000000 * 2^{110}$

$0,10100110 * 2^{010}$  aumento l'esponente a 110 ->  $0,00001010 * 2^{110}$  (2,5)  
 Errore di incolonnamento

Operazione finale:

$0,11000000 * 2^{110}$
$0,00001010 * 2^{110}$
-----
$0,11001010 * 2^{110}$ (50,5)

2)

$$C20 = R25 * F3 + (R25 + R25-30) * F2 + R * P.$$

R25	R25-30	F2	F3	R	P	C20	
1	-		-	1	-	-	1
-	1		1	-	-	-	1
1	-		1	-	-	-	1
-	-		-	-	1	1	1

negli altri casi è 0.

```
/* prende un vettore di interi pari e ne stampa la divisione per due di ciascun elemento */
```

```
#include <stdio.h>
```

```
#define MAX 20
```

```
main()
```

```
{
```

```
    int DIGIT[MAX],risultato[MAX];
```

```
    int i,j;
```

```
    void dividi_per_2(int n[], int m[],int lim);
```

```
    i=0;
```

```
    printf("\n");
```

```
    do
```

```
    {
```

```
        printf("Inserisci il numero %d: ",i);
```

```
        scanf("%d", &DIGIT[i]);
```

```
        i=i+1;
```

```
    }
```

```
    while (DIGIT[i-1]/2*2==DIGIT[i-1]);
```

```
/* i-2 e' la posizione dell'ultimo numero inserito */
```

```
    printf("Il vettore risultato e':\n");
```

```
    dividi_per_2(DIGIT, risultato, i-1);
```

```
    for(j=0;j<i-1;j++)
```

```
        printf("%d ", risultato[j]);
```

```
}
```

```
void dividi_per_2(int n[], int m[], int lim)
```

```
{
```

```
    int i;
```

```
    for(i=0;i<lim;i++)
```

```
        m[i]=n[i]/2;
```

```
}
```

---

**ESAME DI FONDAMENTI DI INFORMATICA I**  
**ESAME DI ELEMENTI DI INFORMATICA**  
**11/06/1997**

**PROVA SCRITTA**

1. Mostrare come viene svolto il calcolo:

$$-18 + 21 + 4.55$$

qualora l'elaboratore adotti per i numeri interi una rappresentazione in complemento a due su 8 bit e per i numeri reali una rappresentazione in virgola mobile con un byte per la mantissa normalizzata e un byte per l'esponente in complemento a due.

Mostrare i passaggi intermedi eseguiti dall'elaboratore con particolare riferimento alle operazioni e alla rappresentazione interna in binario, nonché i relativi eventuali errori compiuti. Scrivere il risultato finale in codifica decimale.

2. Per andare da Bologna a Torino è possibile prendere un aereo e alternativamente una macchina a noleggio o un taxi oppure un treno e un autobus oppure un pullman.

In funzione delle variabili logiche:

- A, con valore 1 se si prende l'aereo;
- MN, con valore 1 se si prende una macchina a noleggio;
- T, con valore 1 se si prende un taxi;
- TR, con valore 1 se si prende un treno;
- BUS, con valore 1 se si prende un autobus;
- P, con valore 1 se si prende un pullman;

Scrivere l'espressione analitica di F e la tabella della verità.

---

**PROVA PRATICA**

Si scriva un programma C (BASIC per chi sostiene l'esame di Elementi di Informatica) che:

- a) legga da terminale una sequenza di interi positivi terminati dal valore 0 (uno su ogni linea) e li inserisca in un vettore DIGIT di 20 componenti (si suppone che la sequenza sia di lunghezza minore o uguale a 20);
- b) Inserisca in un vettore MAGGIORI i numeri della sequenza che sono maggiori del numero precedente nel vettore DIGIT.

Esempio: DIGIT: 6 3 32 76 33 11 13 2

MAGGIORI: 32 76 13

---

SOLUZIONE COMPITO GIUGNO

1)  
 -18    11101110  
 21    00010101  
 -----  
 (1)00000011    riporto

Rappresentazione mantissa + esponente

$4,55_{10} = 100,100011$   
 normalizzo  $0,10010001 * 2^{011}$     Errore di troncamento

Rappresento il risultato dell'operazione sugli interi in floating point.

11.000000:  
 normalizzo  $0,11000000 * 2^{010}$   
 incolonna per sommare con 4,55:  $0,01100000 * 2^{011}$

Operazione:     $0,01100000 * 2^{011} +$   
                   $0,10010001 * 2^{011} =$   
 -----  
                   $0,11110001 * 2^{011} = 7,53125$  invece di 7,55

2)  
 $V = A * (MN + T) + TR * BUS + P$

A	MN	T	TR	BUS	P	V
1	-	1	-	-	-	1
1	1	-	-	-	-	1
-	-	-	1	1	-	1
-	-	-	-	-	1	1



```
/*prende un vettore di interi e inserisci in un vettore MAGGIORI i numeri
maggiori del numero precedente */
```

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

```
main()
{
    int DIGIT[MAX],MAGGIORI[MAX];
    int i,j,max_mag;
    void inserisci_maggiori(int n[], int m[], int lim, int * max_mag);

    i=0;
    printf("\n");
    do
    {
        printf("Inserisci il numero %d: ",i);
        scanf("%d", &DIGIT[i]);
        i=i+1;
    }
    while (DIGIT[i-1]!=0);
    /* i-2 e' la posizione dell'ultimo numero inserito */
    inserisci_maggiori(DIGIT,MAGGIORI, i-1, &max_mag);
    printf("Il vettore risultato e':\n");
    for (j=0;j<max_mag;j++)
        printf("%d ",MAGGIORI[j]);
}
```

```
void inserisci_maggiori(int n[], int m[], int lim, int *j)
{
    int i;
    *j=0;
    for(i=1;i<lim;i++)
        if (n[i]>n[i-1])
        {
            m[*j]=n[i];
            *j=*j+1;
        }
}
```

---

**ESAME DI FONDAMENTI DI INFORMATICA I**  
**ESAME DI ELEMENTI DI INFORMATICA**  
**16/07/1997**  
**PROVA SCRITTA**

1. Mostrare come viene svolto il calcolo:

$$-13 + 27 + 7.8$$

qualora l'elaboratore adotti per i numeri interi una rappresentazione in complemento a due su 8 bit e per i numeri reali una rappresentazione in virgola mobile con un byte per la mantissa normalizzata e un byte per l'esponente in complemento a due.

Mostrare i passaggi intermedi eseguiti dall'elaboratore con particolare riferimento alle operazioni e alla rappresentazione interna in binario, nonché i relativi eventuali errori compiuti. Scrivere il risultato finale in codifica decimale.

2. Per partecipare ad una gara sportiva e' possibile scegliere tra piu' specialita': effettuare una gara podistica oppure una partita di calcio oppure una partita di ping pong e una di pallavolo. In funzione delle variabili logiche:

- GS, con valore 1 se si partecipa alla gara sportiva;
- P, con valore 1 se si effettua una gara podistica;
- C, con valore 1 se si effettua una partita di calcio;
- PP, con valore 1 se si effettua una partita di ping pong;
- PV, con valore 1 se si effettua una partita di pallavolo;

Scrivere l'espressione analitica di GS e la tabella della verità.

---

**PROVA PRATICA**

Si scriva un programma C (BASIC per chi sostiene l'esame di Elementi di Informatica) che:

- a) Legga da terminale una sequenza di caratteri terminati dal carattere '@' (uno su ogni linea) e li inserisca in un vettore CAR di 20 componenti (si suppone che la sequenza sia di lunghezza minore o uguale a 20);
- b) Chiamando una opportuna procedura, inserisca in un secondo vettore di interi, VOCALI, il numero di volte che compare ciascuna vocale nella sequenza (VOCALI[0] per la lettera 'A', VOCALI[1] per 'E', VOCALI[2] per 'I', VOCALI[3] per 'O' e VOCALI[4] per 'U');
- c) Stampi il vettore VOCALI a terminale

Esempio: CAR:    A B A C E I O I I @  
          VOCALI: 2  1  3  1  0

```

#include<stdio.h>
#define MAX 20
void genera_vocali(char CAR[], int vocali[], int lim);
main()
{
    char CAR[MAX];
    int vocali[5];
    int i,j;
    i=0;
    printf("\n");
    do
    {
        scanf("%c", &CAR[i]);
        i=i+1;
    }
    while (CAR[i-1]!='@');
/* i-2 e' la posizione dell'ultimo numero inserito */
    genera_vocali(CAR, vocali, i-1);
    printf("Vocali:\n");
    printf("A: %d\n",vocali[0]);
    printf("E: %d\n",vocali[1]);
    printf("I: %d\n",vocali[2]);
    printf("O: %d\n",vocali[3]);
    printf("U: %d\n",vocali[4]);
}

void genera_vocali(char CAR[], int vocali[], int lim)
{
    int i;
    vocali[0]=0;
    vocali[1]=0;
    vocali[2]=0;
    vocali[3]=0;
    vocali[4]=0;
    for(i=0;i<lim;i++)
        switch (CAR[i])
        {
            case 'A': case 'a':
                vocali[0]=vocali[0]+1;
                break;
            case 'E': case 'e':
                vocali[1]=vocali[1]+1;
                break;
            case 'I': case 'i':
                vocali[2]=vocali[2]+1;
                break;
            case 'O': case 'o':
                vocali[3]=vocali[3]+1;
                break;
            case 'U': case 'u':
                vocali[4]=vocali[4]+1;
        }
}

```

---

**ESAME DI FONDAMENTI DI INFORMATICA I**  
**ESAME DI ELEMENTI DI INFORMATICA**  
**17/10/1997**

**PROVA SCRITTA**

1. Mostrare come viene svolto il calcolo:

$$30 - 17 + 8.4$$

qualora l'elaboratore adotti per i numeri interi una rappresentazione in complemento a due su 8 bit e per i numeri reali una rappresentazione in virgola mobile con un byte per la mantissa normalizzata e un byte per l'esponente in complemento a due.

Mostrare i passaggi intermedi eseguiti dall'elaboratore con particolare riferimento alle operazioni e alla rappresentazione interna in binario, nonché i relativi eventuali errori compiuti. Scrivere il risultato finale in codifica decimale.

2. Per fare la spesa in un grande magazzino è necessario possedere una macchina con un baule capiente, oppure un camioncino oppure disporre di un mezzo pubblico vicino a casa e farsi accompagnare da una persona robusta. In funzione delle variabili logiche:

- S, con valore 1 se si fa la spesa in un grande magazzino;
- M, con valore 1 se si possiede una macchina con un baule capiente;
- C, con valore 1 se si possiede un camioncino;
- PR, con valore 1 se ci si fa accompagnare da una persona robusta;
- MP, con valore 1 se si dispone di un mezzo pubblico vicino a casa;

Scrivere l'espressione analitica di S e la tabella della verità.

---

**PROVA PRATICA**

Si scriva un programma C (BASIC per chi sostiene l'esame di Elementi di Informatica) che:

- a) Legga da terminale una sequenza di numeri interi (uno su ogni linea) terminata da un numero negativo e li inserisca in un vettore INT di 20 componenti (si suppone che la sequenza sia di lunghezza minore o uguale a 20);
- b) Legga da terminale un comando di utente che può essere: `cres` o `decr` che discrimina se l'ordinamento del vettore INT deve essere effettuato in modo crescente o decrescente.
- c) Chiamando una opportuna sub-routine, ordini il vettore INT in un vettore INT-ORD (anch'esso di 20 elementi) in modo crescente se il comando dato dall'utente è `cres` in senso decrescente se il comando dato dall'utente è `decr`;