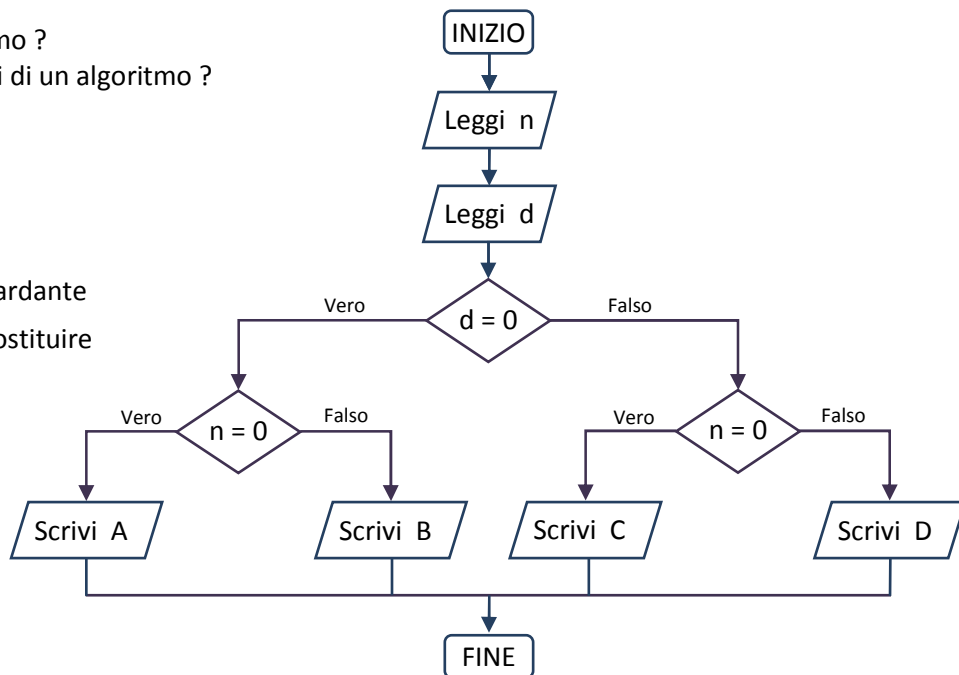


Alunno: _____ Classe: 1 C

1. Cosa si intende con il termine algoritmo ?
2. Quali sono le caratteristiche principali di un algoritmo ?
3. Che cos'è il codice ASCII ?
4. Che cos'è l'UNICODE ?
5. Che cos'è il byte
6. Che cos'è il bit
7. Interpreta il seguente algoritmo, riguardante le frazioni $\frac{n}{d}$, scrivendo le frasi da sostituire al posto delle lettere A, B, C, D, (dove n e d sono due numeri interi).



8. La frase "salve a tutti" è memorizzata utilizzando:
 11 bit 13 bit 13 byte 11 byte
9. In un computer che utilizza parole di 2 byte, MININT = e MAXINT =
10. Effettua i seguenti calcoli nel sistema binario:
 $10010100 - 1110111$ 100101×1011 $1001101010 : 101$
11. Trasforma da un sistema di numerazione ad un altro:
 $(1001101)_2 = ()_{10}$ $(12364)_8 = ()_{10}$ $(A2F)_{16} = ()_{10}$
 $(1001101101)_2 = ()_{16}$ $(69)_{10} = ()_2$
12. In un computer che opera con parole di 2 byte la sequenza a lato corrisponde al numero relativo :
 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1
13. Calcola il complemento a due del numero a lato

0	0	0	1	1	1	0	1	0	0	0	0	0	0	1	1	1
14. In un computer che utilizza 4 byte per memorizzare un numero reale, come viene memorizzato il numero -156000 ?

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32		
15. Spiega come il computer effettua l'operazione +7 - 9.

Valutazione

Esercizio	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Punti	4	4	4	4	4	4	12	4	4	12	20	4	6	8	6
Voto	Punteggio grezzo / 10														

Soluzione

1. Cosa si intende con il termine algoritmo ?

Con il termine algoritmo si intende un procedimento per la risoluzione di un problema, rappresentato in un linguaggio comprensibile all'uomo e adatto ad essere tradotto in un programma eseguibile da un computer.

2. Quali sono le caratteristiche principali di un algoritmo ?

Un algoritmo deve:

- ✚ essere **finito**: la sequenza di istruzioni deve essere finita, deve avere un punto di *Inizio*, dove si avvia l'esecuzione delle azioni, e un punto di *Fine*, dove si interrompe l'esecuzione
- ✚ essere **eseguibile**: le istruzioni devono poter essere eseguite materialmente dall'esecutore
- ✚ essere **non ambiguo**: le istruzioni devono essere interpretate da tutti allo stesso modo
- ✚ essere **generale**: deve essere valido non solo per un particolare problema, ma per una classe di problemi
- ✚ essere **deterministico**: partendo dagli stessi dati iniziali deve portare sempre allo stesso risultato finale indipendentemente dall'esecutore
- ✚ essere **completo**: deve contemplare tutti i casi che si possono verificare durante l'esecuzione

3. Che cos'è il codice ASCII ?

Il codice ASCII (American Standard Code for Information Interchange - Codice Standard Americano per lo Scambio di Informazioni) è uno standard di codifica internazionale dei simboli utilizzati (lettere, numeri, operazioni, segni di interpunzione, ecc.), per poter scambiare dati fra i vari computer del mondo.

4. Che cos'è l'UNICODE ?

L'Unicode è un sistema di codifica per rappresentare univocamente tutti i caratteri dei differenti linguaggi del mondo. Esso assegna un numero univoco ad ogni carattere usato per la scrittura di testi, in maniera indipendente dalla lingua, dalla piattaforma informatica e dal programma utilizzati.

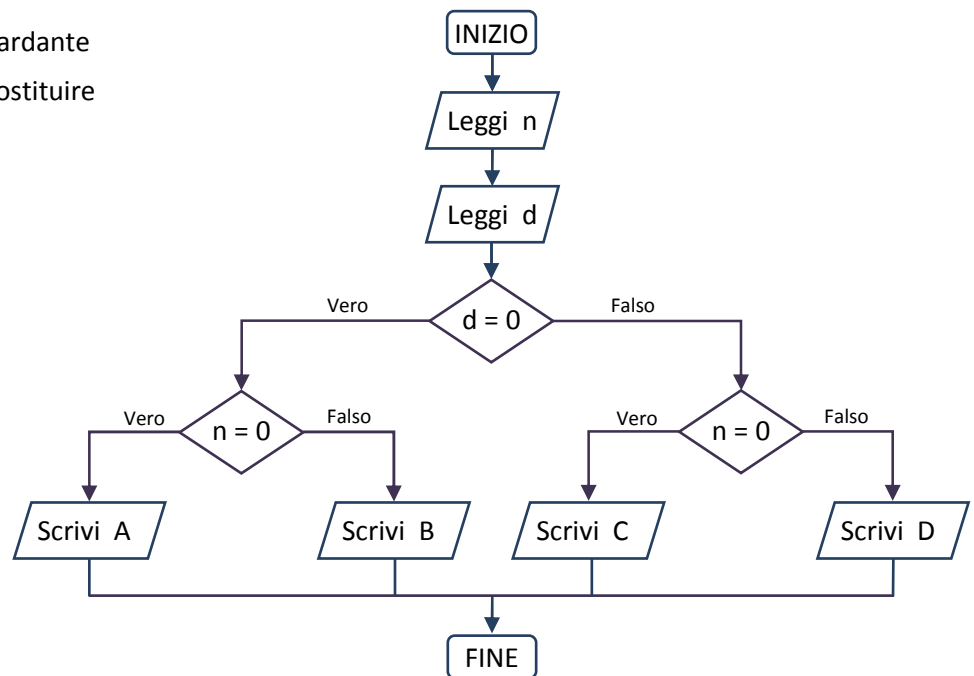
5. Che cos'è il byte

Il byte è una stringa di 8 bit. Il byte rappresenta l'unità minima di memorizzazione di un'informazione.

6. Che cos'è il bit

Il bit è l'unità base d'informazione. Esso può assumere due soli stati: UNO oppure ZERO

7. Interpreta il seguente algoritmo, riguardante le frazioni $\frac{n}{d}$, scrivendo le frasi da sostituire al posto delle lettere A, B, C, D, (dove n e d sono due numeri interi).



- A: $\frac{0}{0}$ Forma indeterminata
 B: $\frac{n \neq 0}{0}$ Forma impossibile
 C: $\frac{0}{d \neq 0} = 0$
 D: $\frac{n \neq 0}{d \neq 0}$ Forma determinata

8. La frase "salve a tutti" è memorizzata utilizzando:

■ 13 byte

9. In un computer che utilizza parole di 2 byte

In un computer che opera con parole di 2 byte (16 bit) la codifica dei numeri interi avviene nel seguente modo: Dei 16 bit a disposizione, il primo a sinistra viene utilizzato per la memorizzazione del segno (0 per il segno positivo e 1 per il segno negativo) mentre gli altri 15 per rappresentare il numero.

0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 Numero **positivo**
 1 1 0 0 1 1 0 0 1 1 1 1 0 1 0 1 Numero **negativo**

Ricordando che con 15 bit si possono ottenere: $2^{15} = 32768$ combinazioni diverse, e facendo corrispondere lo zero alla combinazione di bit: 0000 0000 0000 0000 si ha che i numeri interi assumono valori fra i numeri:

$$\text{MININT} = -2^{15} = -32768 \quad \text{e} \quad \text{MAXINT} = +2^{15} - 1 = +32767$$

10. Effettua i seguenti calcoli nel sistema binario:

$$10010100 - 1110111$$

$$100101 \times 1011$$

$$1001101010 : 101$$

$$\begin{array}{r}
 1 10 10 \\
 0 10 0 0 10 10 \\
 1 0 1 1 0 - \\
 \hline
 1 1 0 1 1 = \\
 \hline
 0 0 1 1 0
 \end{array}$$

$$\begin{array}{r}
 1 0 1 1 \\
 \hline
 1 1 = \\
 \hline
 1 0 1 1 \\
 1 0 1 1 - \\
 \hline
 0 0 0 0 - \\
 1 0 1 1 - \\
 \hline
 1 0 1 1 1
 \end{array}$$

$$\begin{array}{r}
 1 0 1 0 0 0 \\
 \hline
 1 1 \\
 \hline
 1 0 1 \\
 0 \\
 \hline
 1 0 \\
 0 \\
 \hline
 1 1 \\
 0 \\
 \hline
 1 0 1 \\
 0 \\
 \hline
 1 0 \\
 0 \\
 \hline
 1
 \end{array}
 \quad \left| \begin{array}{r}
 1 1 \\
 \hline
 1 1 1 1
 \end{array}
 \right.$$

11. Trasforma da un sistema di numerazione ad un altro:

$$\begin{aligned}
 (1001101)_2 &= (1 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0)_{10} = \\
 &= (64 + 0 + 0 + 8 + 4 + 0 + 1)_{10} = (77)_{10}
 \end{aligned}$$

$$\begin{aligned}
 (12364)_8 &= (1 \cdot 8^4 + 2 \cdot 8^3 + 3 \cdot 8^2 + 6 \cdot 8^1 + 4 \cdot 8^0)_{10} = \\
 &= (4096 + 1024 + 192 + 48 + 4)_{10} = (5364)_{10}
 \end{aligned}$$

$$(A2F)_{16} = (A \cdot 16^2 + 2 \cdot 16^1 + F \cdot 16^0)_{10} = (10 \cdot 256 + 2 \cdot 16 + 15 \cdot 1)_{10} = (2560 + 32 + 15)_{10} = (2607)_{10}$$

$$(1001101101)_2 = (10.0110.1101)_2 = (26D)_{16}$$

$$(69)_{10} = (1000101)_2$$

69	1
34	0
17	1
8	0
4	0
2	0
1	1
0	

12. In un computer che opera con parole di 2 byte la sequenza a lato corrisponde al numero relativo :

1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1

La prima cifra indica il segno 1 = -

$$(1011)_2 = (1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0)_{10} = (8 + 2 + 1)_{10} = (11)_{10}.$$

Pertanto la sequenza delle cifre della traccia corrisponde al numero -11 .

13. Calcola il complemento a due del numero a lato

0	0	0	1	1	1	0	1	0	0	0	0	0	1	1	1	+ = =
1	1	1	0	0	0	1	0	1	1	1	1	1	0	0	0	
															1	
1	1	1	0	0	0	1	0	1	1	1	1	1	0	0	1	

14. In un computer che utilizza 4 byte per memorizzare un numero reale, come viene memorizzato il numero -156000 ?

Rappresentazione in virgola fissa

Nella rappresentazione in virgola fissa, la virgola ha sempre la stessa posizione. Alla parte intera del numero è riservato sempre lo stesso numero di bit, come pure alla parte decimale.

Utilizzando 4 byte = 32 bit una possibile rappresentazione del numero reale in virgola fissa potrebbe essere la seguente

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
Segno	Parte intera																							Parte decimale							

↑
Posizione della virgola

Con questo tipo di rappresentazione il numero -156000 è codificato nel seguente modo:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
1	0	0	0	0	0	1	0	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0

Rappresentazione in virgola mobile

Essendo il numero negativo, il primo bit è uno.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	
1	esponente								mantissa																							

b. determinare la rappresentazione in base 2 del valore assoluto del numero

La rappresentazione in base 2 è: $(156000)_{10} = (100110000101100000)_2$

$$(156000)_{10} = (1 \cdot 2^{17} + 1 \cdot 2^{14} + 1 \cdot 2^{13} + 1 \cdot 2^8 + 1 \cdot 2^6 + 1 \cdot 2^5)_{10} = (100110000101100000)_2$$

c. Normalizzare il numero

Il numero normalizzato si ottiene spostando la virgola subito dopo il primo bit.

Il numero normalizzato è: $(1,00110000101100000 \cdot 2^{17})_2$

Si riscrive solo la mantissa 00110000101100000 a partire dal decimo bit.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
1									0	0	1	1	0	0	0	0	1	0	1	1	0	0	0	0							

c. Polarizzare l'esponente

L'esponente polarizzato si ottiene aggiungendo all'esponente originario 17 il numero fisso 127, detto bias.

Pertanto l'esponente polarizzato è: $(17 + 127)_{10} = (144)_{10} = (1 \cdot 2^7 + 1 \cdot 2^4)_2 = (10010000)_2$

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
1	1	0	0	1	0	0	0	0	1	1	0	1	1	0	1	0	1														

d. Completare la mantissa con degli zeri finali

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
1	1	0	0	1	0	0	0	0	1	1	0	1	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15. Spiega come il computer effettua l'operazione $+7 - 9$.

1. Il computer trasforma la differenza in una somma: $+7 - 9 = +7 + (-9)$
2. In seguito, con la regola del complemento a due, calcola l'opposto di $+9$, cioè: -9 .
3. infine effettua la somma di $+7$ e -9