

Eserciziario per il corso sul linguaggio C

Rel. 1.0

30.8.2015

Luigi Ferrari

Indice

| | |
|------------------------------------------------------------------------|-----------|
| 1. Prerequisiti e flowchart..... | 2 |
| 1.1. Prerequisiti e flowchart..... | 2 |
| 2. Basi linguaggio C..... | 5 |
| 2.1. Tipi di dato..... | 5 |
| 3. Funzioni..... | 7 |
| 3.1. Analisi, sintassi e prototipi..... | 7 |
| 3.2. Algoritmi sequenziali, input ed output..... | 8 |
| 3.3. Passaggio di argomenti modificabili (passaggio di puntatori)..... | 9 |
| 4. Controllo di flusso..... | 10 |
| 4.1. Test..... | 10 |
| 4.2. Algoritmi iterativi (cicli)..... | 11 |
| 5. Array e stringhe..... | 18 |
| 5.1. Array monodimensionali..... | 18 |
| 5.1.1. Algoritmi di ordinamento..... | 19 |
| 5.2. Stringhe..... | 21 |
| 5.3. Array multidimensionali..... | 22 |
| 6. Strutture Dati..... | 25 |
| 6.1. Struct e typedef..... | 25 |
| 6.2. Allocazione dinamica..... | 25 |
| 6.3. Strutture dati complesse..... | 25 |
| 6.4. Esercizi di riepilogo..... | 26 |
| 7. Files..... | 27 |
| 7.1. Esercizi di base..... | 27 |
| 7.2. Esercizi complessi..... | 28 |

Licenza

Quest'opera è stata rilasciata con licenza Creative Commons Attribuzione - Condividi allo stesso modo 3.0 Unported. Per leggere una copia della licenza visita il *sito web* (<http://creativecommons.org/licenses/by-sa/3.0/deed.it>) o spedisce una lettera a Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.

Il testo completo della licenza sul sito di Creative Commons, <http://creativecommons.org/licenses/by-nc-sa/3.0/legalcode>.



Corso di OOP by Luigi Ferrari is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported (CC BY-NC-SA 3.0).

Tu sei libero:

- di riprodurre, distribuire, comunicare al pubblico, esporre in pubblico, rappresentare, eseguire e recitare quest'opera
- di modificare quest'opera

alle seguenti condizioni:

- **Attribuzione:** Devi attribuire la paternità dell'opera nei modi indicati dall'autore o da chi ti ha dato l'opera in licenza e in modo tale da non suggerire che essi avallino te o il modo in cui tu usi l'opera.
- **Condividi allo stesso modo:** Se alteri o trasformi quest'opera, o se la usi per crearne un'altra, puoi distribuire l'opera risultante solo con una licenza identica o equivalente a questa.
- **Non commerciale:** Non puoi usare quest'opera per fini commerciali.

Prendendo atto che:

- **Rinuncia:** E' possibile rinunciare a qualunque delle condizioni sopra descritte se ottieni l'autorizzazione dal detentore dei diritti.
- **Pubblico Dominio:** Nel caso in cui l'opera o qualunque delle sue componenti siano nel pubblico dominio secondo la legge vigente, tale condizione non è in alcun modo modificata dalla licenza.

Altri Diritti

La licenza non ha effetto in nessun modo sui seguenti diritti:

- Le eccezioni, libere utilizzazioni e le altre utilizzazioni consentite dalla legge sul diritto d'autore;
- I diritti morali dell'autore;
- Diritti che altre persone possono avere sia sull'opera stessa che su come l'opera viene utilizzata, come il diritto all'immagine o alla tutela dei dati personali.

Nota

Ogni volta che usi o distribuischi quest'opera, devi farlo secondo i termini di questa licenza, che va comunicata con chiarezza.

1. Prerequisiti e flowchart

1.1. Prerequisiti e flowchart

1. Rappresenta graficamente con schemi di iterazione o di selezione le seguenti frasi

- 1) Se fa caldo
 allora apri la finestra
- 2) Finché fa caldo
 lascia la finestra aperta
- 3) Finché ci sono fogli nel cassetto
 prendi un foglio
- 4) Se ci sono quaderni nel cassetto
 allora prendi un quaderno
 altrimenti vai a comprarne uno
- 5) Ripeti
 lava un piatto
 finché i piatti non sono tutti puliti
- 6) Finché c'è un piatto sporco
 lava un piatto
- 7) Finché hai un voto insufficiente
 studia per recuperare
- 8) Ripeti
 studia di più
 finché non prendi un voto sufficiente
- 9) Finché non hai capito cosa è l'iterazione
 fai un esercizio sull'iterazione
- 10) Se ho capito la spiegazione
 riordino gli appunti
 altrimenti chiedo chiarimenti
- 11) Se c'è spazio nella scheda di memoria del cellulare
 finché non finisce
 scatto fotografie
 altrimenti compro delle cartoline

2. Rappresentare con un flow-chart i seguenti procedimenti.
- 1) Travasare il vino contenuto in una damigiana da 20 litri in bottiglie da 3/4.
 - 2) Registrare due CD da 60 minuti su cassette da 45 minuti.
 - 3) Eseguire la somma tra due numeri di due cifre ciascuno.
 - 4) Eseguire la differenza tra due numeri di due cifre ciascuno.
 - 5) Eseguire la moltiplicazione tra due numeri di due cifre ciascuno.
 - 6) Eseguire la divisione tra due numeri di due cifre ciascuno.
 - 7) Calcolare la lunghezza dell'ipotenusa di un triangolo rettangolo di cui si conoscono i cateti.

- 8) Prepararsi e fare una verifica di elettronica.
 - 9) Fare il flow chart di un procedimento.
 - 10) Vincere al totocalcio.
3. **Orario:** Descrivere mediante un diagramma di flusso un programma in grado di calcolare quanti secondi sono passati tra la mezzanotte e un'ora indicata mediante ora (intero), minuti (intero), secondi (intero) e un numero che rappresenta l'essere di mattino o di pomeriggio (0 per il mattino e 1 per il pomeriggio).
 4. **Cedolino:** Descrivere mediante un diagramma di flusso un programma in grado di stampare il ruolino di paga settimanale di un lavoratore inglese.
 Allo stipendio lordo va dedotto il 5% per la liquidazione, il 25% di ciò che rimane come imposta sul reddito e una deduzione fissa di 14.50 sterline alla settimana per l'assistenza medica.
 Nel cedolino vanno indicati il lordo, il netto e tutte le deduzioni previste.
 5. **Anni bisestili:** Descrivere mediante un diagramma di flusso un programma in grado di calcolare se un certo anno è bisestile o no. Un anno è bisestile se il suo numero è divisibile per 4, con l'eccezione degli anni secolari (quelli divisibili per 100) che non sono divisibili per 400.
 Suggerimento: utilizzare il resto della divisione intera.
 6. **Somma di ore:** Descrivere mediante un diagramma di flusso un programma in grado di fare la somma tra due ore, presentate nella forma hh:mm:ss.
 Il programma deve tener conto che se si superano le ore 23:59:59 si deve ripartire da 00:00:00.
 7. **Rappresentazione dei numeri nel sistema binario:** Copiare sul quaderno e completare:

```

Il più grande numero intero positivo rappresentabile con

4  cifre binarie è .....      10 cifre binarie è .....
16 cifre binarie è .....      20 cifre binarie è .....

Convertire in decimale 1 0 1 1 0 0 1 0 .....
Convertire in binario  120 .....

convertire da decimale a binario (compl. a 2, 8 bit) i numeri
-20 .....
-100 .....

eseguire la moltiplicazione dei due numeri binari      1 1 1 0 0 x
                                                         1 1 0
                                                         -----
    
```

8. **Numerazione esadecimale:** Tradurre in binario e in esadecimale, sul quaderno, i seguenti valori interi espressi in base decimale: 13, 56 208, 120, -4, -1, -100. Utilizzare 8 bit per ogni valore.
9. **Numerazione esadecimale:** Trasformare, sul quaderno, in binario e in decimale i seguenti numeri esadecimali che rappresentano valori interi con 8 bit: 30, 4A, D9, FF
10. **Calcolo del resto:** Scrivere il flowchart e fare la trace-table di almeno due casi significativi del seguente algoritmo:

```
1. prendi due numeri A e B
2. se A non è maggiore di B scambiali
3. toglì B da A
4. se A è positivo torna a 3.
5. se A è diverso da zero aggiungigli B
6. stampa A
```

11. **Potenza:** Scrivere il flowchart e fare la trace-table di almeno due casi significativi del seguente algoritmo:

```
1. prendi due numeri A e B
2. assegna a C il valore 1
3. assegna a C il valore di C per A
4. decrementa B
4. se A è diverso da zero torna a 3.
6. stampa C
```

12. **Radice quadrata:** Scrivere il flowchart e fare la trace-table di almeno due casi significativi del seguente algoritmo:

```
1. prendi un numero N
2. assegna a R il valore 1
3. assegna a X il valore 1
4. calcola D come differenza tra R*R e N
5. se D/N e' inferiore a 0,01 vai a 12.
6. se D e' maggiore di zero
7. toglì X a R
8. vai a 10.
9. assegna a X il valore di X/10
10. aggiungi X a R
11. vai a 3.
12. stampa R
```

2. Basi linguaggio C

2.1. Tipi di dato

1. **Tipi di dato:** Utilizzando le risorse a disposizione, completare la tabella (prof. Mario Janin, 1996).

Numeri interi

| interi | char | int | long | unsigned | unsigned long |
|--------------------------|------|-----|------|----------|---------------|
| bytes occupati | | | | | |
| valore max | | | | | |
| valore min | | | | | |
| specificatore di formato | | | | | |

Numeri reali

| virgola mobile | float | double | long double |
|--------------------------|-------|--------|-------------|
| bytes occupati | | | |
| valore max | | | |
| valore min | | | |
| specificatore di formato | | | |

2. Scrivere un piccolo programma che dichiari delle variabili: una intera, una reale, una di tipo carattere. Stamparne il valore di ognuna (una per riga). Il programma assegna POI ad ognuna di loro un valore e ripete la stampa.

Infine, aggiunge 2 ad ognuna delle tra variabile e ne stampa di nuovo il valore. Fare delle osservazioni al riguardo.

3. Scrivere un programma che dichiara due variabili a e b, chiede all'utente di inserire due valori e stampa la somma, la differenza, il prodotto e il risultato della divisione.
Fare diverse prove per verificare che funzioni sempre correttamente.

3. Funzioni

ATTENZIONE

Per ogni funzione che viene richiesta devono essere eseguiti i seguenti passi, che comprendono l'analisi, la codifica e il test.

1. la scelta del nome
2. la comprensione di quanto viene richiesto
3. l'identificazione del numero, tipo e significato di ognuno dei parametri in ingresso (se presenti)
4. l'identificazione del tipo e del significato del valore di ritorno (se presente)
5. la scrittura di un commento con i quattro punti precedenti
6. la dichiarazione con il prototipo
7. la definizione della funzione (il codice)
8. il main di prova della funzione (se possibile automatico, cioè che non richiede input all'utente)

3.1. Analisi, sintassi e prototipi

1. Somma di due numeri reali
2. Calcolo prezzo scontato di un articolo



3. Calcolo area di un triangolo rettangolo
4. Calcolo area di un triangolo dati i lati (ma è davvero un triangolo?)
5. Conversione di un importo in Euro nel corrispondente in dollari
6. Le azioni dell'azienda Peia passano da x Euro a y Euro nel corso di un anno. Calcolare il guadagno o la perdita percentuale.
7. Dato il tempo sui 100 metri di un atleta calcolare la velocità media in km/h.



8. Dato un carattere mostrare sul video il carattere stesso e il suo codice ASCII in tre forati: in decimale, in ottale ed in esadecimale.

9. Data una distanza e il tempo impiegato per percorrerla, trovare la velocità media.
10. **Piastrellista:** Realizzare una funzione che date le misure del pavimento di una stanza rettangolare, la dimensione delle piastrelle e il numero di piastrelle di una scatola trova il numero di scatole di piastrelle necessario per coprirlo tutto.



3.2. Algoritmi sequenziali, input ed output

1. **On the road:** Un ragazzo fa un viaggio in moto, partendo con il serbatoio pieno. Durante il viaggio si ferma più volte a fare benzina e ogni volta mette nel serbatoio 8 litri. Alla fine del viaggio rifà il pieno (in modo da avere il serbatoio come alla partenza) e prende nota dei chilometri percorsi (il contachilometri non riporta le frazioni).

Scrivere una funzione che calcoli il consumo medio di carburante espresso in chilometri per ogni litro.



Immagine tratta dal film "I diari della motocicletta"

2. **Preventivo:** Scrivere un programma che consenta ad una impresa edile di fare dei preventivi di spesa sui lavori. I dati che devono essere forniti al programma sono il costo del materiale, il numero di giorni di lavoro previsti e il numero di operai da dedicarvi.

Una giornata di un operaio costa 120 Euro. Al totale deve essere aggiunta una quota pari al 4% di assicurazione. Sul totale complessivo deve essere calcolata l'IVA al 20%.

Alla fine deve essere visualizzato un prospetto il più possibile simile al seguente:

Impresa Massacan
Via Muratori 99
16100 Genova

Preventivo

Costo materiali

1500,00

| | | | |
|-------------------------------------|-----------|---------|------|
| Mano d'opera 3 operai per 10 giorni | | 3600,00 | |
| | | ----- | |
| | SubTotale | 5100,00 | Euro |
| Assicurazione (4%) | | 204,00 | Euro |
| | | ----- | |
| | SubTotale | 5304,00 | Euro |
| IVA (20%) | | 1060,80 | Euro |
| | | ----- | |
| | Totale | 6464,80 | Euro |

3.3. Passaggio di argomenti modificabili (passaggio di puntatori)

1. Scrivere una funzione che riceve l'indirizzo di una variabile intera e ne incrementa il valore contenuto.
2. Scrivere una funzione che scambia il contenuto di due variabili reali.
3. Scrivere una funzione che visualizza l'indirizzo e il valore contenuto in una variabile reale.
4. Scrivere una funzione che riceve l'indirizzo di una variabile intera x e un valore intero n; la funzione modifica il valore di x aggiungendogli n e restituisce e restituisce il valore della differenza tra x e n.
5. Data una distanza e il tempo impiegato per percorrerla, trovare la velocità media in km/h e in m/s.

4. Controllo di flusso

ATTENZIONE

Per ogni funzione che viene richiesta devono essere eseguiti i seguenti passi, che comprendono l'analisi, la codifica e il test.

1. la scelta del nome
2. la comprensione di quanto viene richiesto
3. l'identificazione del numero, tipo e significato di ognuno dei parametri in ingresso (se presenti)
4. l'identificazione del tipo e del significato del valore di ritorno (se presente)
5. la scrittura di un commento con i quattro punti precedenti
6. la dichiarazione con il prototipo
7. la definizione della funzione (il codice)
8. il main di prova della funzione (se possibile automatico, cioè che non richiede input all'utente)

4.1. Test

1. Dati tre numeri restituire il massimo.
2. Dati tre numeri visualizzarli in ordine.
3. Dati tre numeri scabiarne il contenuto in modo che il primo contenga in valore minore, il secondo quello intermedio e l'ultimo il valore massimo.
4. **Controllo input:** Dati un valore minimo e un valore massimo chiedere un numero reale in input all'utente e continuare a chiederlo fino a che non viene inserito un numero compreso tra i due valori. Restituisce il valore scelto.
5. **Somma di angoli:** sommare due angoli espressi in gradi, primi e secondi. Per angoli maggiori di 360 gradi occorre ricondursi a valori minori togliendo un opportuno multiplo di 360 gradi.
6. Indicare se un numero intero e' divisibile per un altro.
7. Quale tra tre numeri e' il piu' piccolo.
8. Mostrare tre numeri in ordine crescente.
9. Somma di due ore espresse in ore, minuti e secondi.

Esempio

```

h  m  s
22 14 30 +
 3 50 27 =
-----
2 04 57

```

Il programma deve tener conto che se si superano le ore 23h59m59s si deve ripartire da 0h0m0s.(h sta per ore, m per minuti, s per secondi).

10. **Vinaio:** Un vinaio propone uno sconto a chi compra una certa quantità minima di vino. Si visualizza quanto l'acquirente spende, in base alla quantità ed al prezzo non scontato di un litro di vino.
11. **Funzioni trigonometriche:** dato tipo di funzione (seno, coseno, tangente) e il valore dell'angolo in radianti, calcola se la funzione e' positiva in quel punto.
Nota: utilizzare le funzioni matematiche del linguaggio C (da cercare sulle risorse a disposizione).

4.2. Algoritmi iterativi (cicli)

1. Visualizzare su video un quadrato di lato N tutto pieno di asterischi tranne che sulla diagonale principale. Esempio con N=10:

```

*****
*  *****
** *****
*** *****
**** *****
***** *****
*****  ****
*****   ***
*****    **
*****     *
*****

```

2. **Tabella dei codici ASCII:** Stampare a video la tabella dei codici ASCII, escludendo i primi 32 caratteri (codici ASCII da 0 a 31).
3. **Inserimento di una sequenza:** richiedere l'introduzione di più numeri interi fino a che non viene introdotto il valore 0. A quel punto deve comunicare quanti numeri positivi e quanti negativi sono stati introdotti.
4. richiedere l'introduzione di più numeri interi fino a che non sono più in sequenza strettamente crescente. Mostra la somma dei valori crescenti introdotti.
5. **Calcolo della media:** Forniti dall'utente un numero N e N numeri reali calcolare la media
6. **Calcolo del Fattoriale:** Dato N, numero intero, calcolare il fattoriale ($4! = 1*2*3*4 = 24$)
7. **Prodotto:** Dati due numeri interi calcolarne il prodotto mediante successive somme. Ottimizzare l'algoritmo.
8. **Deposito di capitale dopo N anni:**
Indicare, in assenza di prelievi e versamenti l'importo dopo N anni con tasso dato e interessi reinvestiti.
9. **Raddoppio del capitale investito:**
Deposito di capitale: calcolare quanti anni sono necessari per raddoppiare il capitale iniziale con tasso dato e interessi reinvestiti.

10. **Tabella di conversione della temperatura:** creare una tabella di conversione tra le varie unita' di misura della temperatura (Celsius, Farheneit e Kelvin). La tabella deve prendere in considerazione valori di temperatura in Celsius compresi tra un Tmin e un Tmax. I successivi valori della temperatura a partire dall'estremo inferiore Tmin si ottengono incrementando ogni volta, con passo predefinito deltaT, il valore della temperatura fino al valore dell'estremo superiore Tmax.
11. **Durata di un mutuo:** Per l'acquisto di una casa si vuole chiedere un mutuo di 100.000 Euro. La banca concede il mutuo all'interesse del 4% e richiede il versamento di una quota trimestrale di 2.000 Euro. Si calcoli in quanto tempo il mutuo verra' estinto.
- Osservazione. Sapendo che l'interesse annuo richiesto dalla banca è del 4% si ragiona nel seguente modo: ogni anno si deve pagare l'interesse sul capitale residuo, più una quota di capitale per ridurre il debito. Dato che le quote annue pagate sono costanti, risulta che nei primi anni le quote sono composte soprattutto di interessi mentre negli ultimi, in cui l'interesse è più basso perchè la maggior parte del capitale è stato restituito, le quote sono composte soprattutto di capitale da restituire.
12. **Divisori di un numero:** Dato un numero N intero positivo trovare i suoi divisori.
13. **Tabella di dati cinematici:** Costruire una tabella che, secondo per secondo, fornisca i dati di posizione e velocità relativi al moto di un carrello trascinato su una rotaia orizzontale dalla caduta libera di un corpo collegato al carrello tramite una fune e una carrucola posta su una estremità della rotaia. La rotaia è del tipo a cuscino d'aria e quindi l'attrito è pressochè nullo. Dati: Massa_corpo=0.4 Kg, Lunghezza_rotaia=2m.
- Si ricorda: $F=ma$, $a=\text{variazione di velocità}/\text{variazione di tempo}$.
14. **Massimo comun divisore:** Dati due numeri interi positivi calcolare il MCD.

| | | | | |
|----|---|-----|---|-------------------------------------------|
| 36 | 2 | 120 | 2 | $36 = 2^2 \times 3^2$ |
| 18 | 2 | 60 | 2 | $120 = 2^3 \times 3 \times 5$ |
| 9 | 3 | 30 | 2 | |
| 3 | 3 | 15 | 3 | $\text{MCD}(36, 120) = 2^2 \times 3 = 12$ |
| 1 | | 5 | 5 | |
| | | 1 | | |

15. **Minimo comune multiplo:** Dati due numeri interi positivi calcolare il mcm.

| | | | |
|-----------------------------------------------------------------------------------|---|--------------------|---|
| 30 | 2 | 12 | 2 |
| 15 | 3 | 6 | 2 |
| 5 | 5 | 3 | 3 |
| 1 | | 1 | |
| $30 = 2 \cdot 3 \cdot 5$ | | $12 = 2^2 \cdot 3$ | |
| <u>m.c.m.</u> $(30 \text{ y } 12) = 2^2 \cdot 3 \cdot 5 = 4 \cdot 3 \cdot 5 = 60$ | | | |

16. **Calcolo del seno per serie:** Il valore del seno di un angolo (espresso in radianti) può essere ottenuto con la formula seguente (^ = elevamento a potenza):

$$\sin(x) = x - x^3/3! + x^5/5! - \dots (-1)^n x^{(2n+1)}/(2n+1)! + \dots$$

Calcolare quanti termini sono necessari per ottenere una approssimazione al valore di $\sin(x)$ pari al P per cento.

Suggerimento: per ogni valore ottenuto si calcola il margine di errore: se il valore ottenuto è diverso dal precedente per meno del margine ci si ferma.

17. **Espansione termica:** La lunghezza L di una barra metallica ad una temperatura T (espressa in gradi centigradi) è data dall'equazione:

$$L = L_0 + E \cdot T \cdot L_0$$

dove L_0 è la lunghezza a 0 gradi ed E è il coefficiente di espansione.

Produrre una tabella che riporti la lunghezza di una barra a diverse temperature comprese tra 0 e 100 gradi, supponendo che, alla temperatura di 20 gradi, la barra sia lunga esattamente 1 metro.

Il programma deve essere in grado di acquisire il coefficiente di espansione della barra da prendere in considerazione.

18. **Tabella di Pitagora:** Stampare la tabellina pitagorica.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|----|----|----|----|----|----|----|----|----|----|-----|
| 1 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 2 | 2 | 4 | 6 | 8 | 10 | 12 | 14 | 16 | 18 | 20 |
| 3 | 3 | 6 | 9 | 12 | 15 | 18 | 21 | 24 | 27 | 30 |
| 4 | 4 | 8 | 12 | 16 | 20 | 24 | 28 | 32 | 36 | 40 |
| 5 | 5 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 | 50 |
| 6 | 6 | 12 | 18 | 24 | 30 | 36 | 42 | 48 | 54 | 60 |
| 7 | 7 | 14 | 21 | 28 | 35 | 42 | 49 | 56 | 63 | 70 |
| 8 | 8 | 16 | 24 | 32 | 40 | 48 | 56 | 64 | 72 | 80 |
| 9 | 9 | 18 | 27 | 36 | 45 | 54 | 63 | 72 | 81 | 90 |
| 10 | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 |

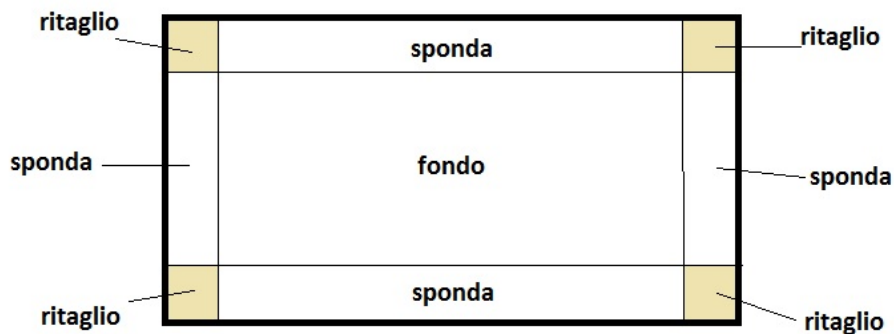
19. **Successione di Fibonacci:** Calcolare i primi N numeri della successione di Fibonacci. La successione di Fibonacci è quella in cui il numero successivo si ottiene per somma dei due elementi precedenti. La successione deve essere inizializzata con i numeri 1 e 1.

Esempio: 1 1 2 3 5 8 13 21 34 55

20. **La vasca - ricerca del massimo volume con superficie limitata:** Per realizzare un esperimento di idraulica in laboratorio è necessario costruire un serbatoio. In magazzino sono disponibili alcuni fogli rettangolari di lamierino di ottone, di misure diverse. Bisogna poter valutare velocemente il massimo volume ottenibile con i diversi fogli, pensando di realizzare un serbatoio a forma di parallelepipedo, scoperto. Chi ha analizzato il problema

ha già scartato l'ipotesi di risolverlo studiando una funzione di terzo grado quale è, in questo caso, il volume del prisma rispetto a uno spigolo.

Definire l'algoritmo per acquisire le misure di un foglio e produrre per via enumerativa un valore approssimato del massimo volume ottenibile, pensando di piegare il foglio come illustrato:

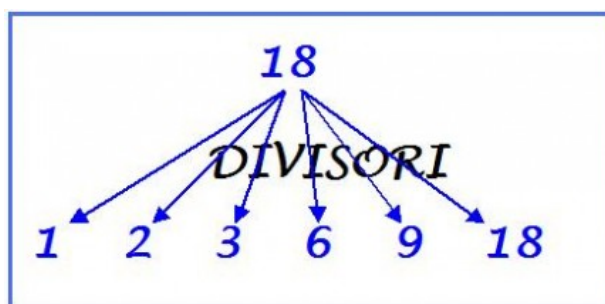


21. **Allevamento bovino:** Un allevamento bovino è costituito da un numero iniziale di 1500 capi. Di essi il 67% sono mucche e l'88% delle mucche dà alla luce nel corso dell'anno un vitello. Il 16% dei capi viene macellato o muore ogni anno per cause naturali. Produrre una tabella che mostri l'andamento dell'allevamento nei prossimi 10 anni.
22. **Calcolatrice da tavolo:** Realizzare una calcolatrice da tavolo: chiede un operatore (+, -, *, /) e due operandi e mostra il risultato dell'operazione. Il programma deve consentire l'esecuzione di un qualsiasi numero di operazioni consecutive senza uscire.
23. **Conversioni di istruzioni di ciclo:** Dato il seguente algoritmo disegnare il flow-chart e riscriverlo trasformando i cicli for in while e in do..while.

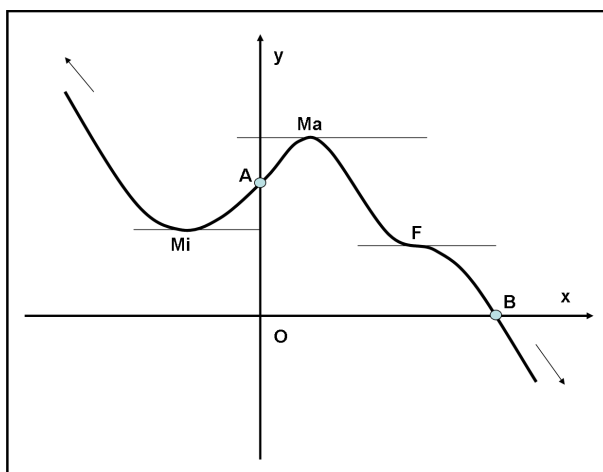
```
void mistero(){
    int conta;
    char lettera;
    for (conta=1; conta<=5; conta++){
        printf("%d ", conta);
        for(lettera = 'A'; lettera <= 'Z'; lettera++)
            printf("%c", lettera);
        printf("\n");
    }
}

void main(){
    mistero();
    getch();
}
```


24. **Quadratura del cerchio:** Determinare per tentativi la dimensione del lato di un quadrato che abbia la stessa area di un cerchio di raggio dato. Non si deve utilizzare la funzione sqrt().
25. **Minimo numero reale:** Trovare per tentativi il più piccolo numero reale rappresentabile in linguaggio C. (suggerimento: x è troppo piccolo se $1+x=1$; La precisione viene determinata decidendo come far decrescere x).
26. **Primalità:** Determinare se un numero intero è primo. Note: il valore 1 non è primo.
27. **Divisori consecutivi:** Trovare tutti i numeri inferiori a 10000 che hanno come divisori tre numeri consecutivi. La visualizzazione deve essere in ordine crescente.



28. **Tabelle di funzioni trigonometriche:** Visualizzare a scelta, anche più volte, le tabelle delle funzioni $\sin(x)$, $\cos(x)$ e $\tan(x)$ con passo di un grado e nell'intervallo di un periodo.
29. **Tabelle di funzioni:** dati i valori di a , b e c reali e x_{\min} e x_{\max} sempre reali, visualizzare una tabella con due colonne, x e $f(x)$ dove $f(x) = a \cdot x^2 + b \cdot x + c$. Il valore x parte da x_{\min} e arriva a x_{\max} in modo che nella tabella ci siano 20 valori diversi di x .



30. **switch case:** Scrivere una funzione che prende in input un numero intero e una lettera e fornisce le seguenti informazioni sullo schermo a seconda della lettera: a) e' positivo o no; b) è pari o dispari; c) è divisibile per 3; d) è divisibile per cinque; e) è divisibile per 7. Restituisce la lettera maiuscola della scelta se la risposta è positiva, la lettera minuscola se la risposta è negativa. Se la lettera non è compresa tra quelle mostra a video "input errato, la lettera x non è valida."
31. **Gioco di Nim:** Si tratta di un gioco ben conosciuto, con un certo numero di varianti. Utilizzeremo una versione che ha una strategia interessante per arrivare alla vittoria. Due

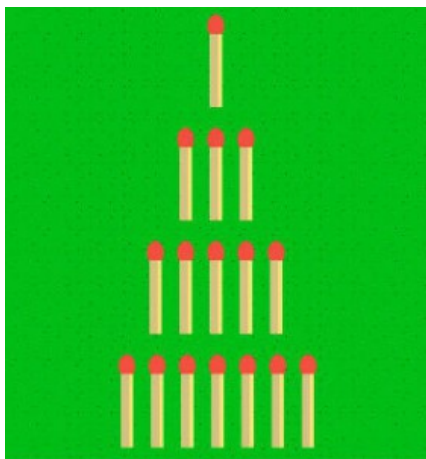
giocatori prendono a turno le biglie da un mucchio. In ciascun turno, il giocatore sceglie quante biglie levare: deve prenderne almeno una, ma non oltre la metà del mucchio. Quindi tocca all'altro giocatore. Perde chi rimane con l'ultima biglia.

Scrivete un programma completo in cui il computer gioca contro un avversario umano. Generate un numero intero casuale, compreso fra 10 e 100, per indicare il numero iniziale di biglie. Generate un altro numero intero casuale, compreso fra zero e uno, per decidere se la prima mossa tocca al computer o al giocatore. Generate un numero intero casuale, compreso fra zero e uno, per stabilire se il computer giocherà in modo intelligente o stupido. Nel modo stupido, quando tocca il suo turno, il computer si limita a sottrarre dal mucchio un numero casuale di biglie, purché sia una quantità ammessa (compresa fra 1 e $n/2$). Nel modo intelligente, il computer leva il numero di biglie sufficiente affinché il numero di quelle rimanenti sia uguale a una potenza di due, meno uno, ovvero 3, 7, 15, 31 o 63. E' sempre una mossa valida, eccetto quando il numero delle biglie è inferiore di un'unità a una potenza di due. In questo caso, il computer preleverà una quantità casuale, purché ammessa.

Noterete che, nel modo intelligente, il computer non si può battere quando ha la prima mossa, a meno che il mucchio non contenga 15, 31 o 63 biglie. Naturalmente, un giocatore umano che ha la prima mossa, e che conosca la strategia vincente, può vincere contro il computer.

(Horstmann, P6.17).

32. **Gioco dei fiammiferi:** Sul tavolo sono presenti 11 fiammiferi. Ogni giocatore a turno ne può prendere 1, 2 o 3 a sua scelta. Perde chi prende l'ultimo. Trovare la strategia vincente e realizzare un algoritmo che la applichi giocando con voi.
33. **Gioco di Marienbad:** Simile al gioco di Nim, si gioca con i fiammiferi.



Le regole del gioco sono le seguenti:

- I due giocatori muovono a turno, togliendo fiammiferi dalla tavola.
- A chi tocchi la prima mossa si decide per sorteggio, o per accordo.
- Si possono prendere tanti fiammiferi quanti si vogliono (almeno uno) ma da una sola riga.
- Chi resta con l'ultimo fiammifero ha perso.

Supponiamo che la partita si giochi tra due giocatori "perfetti", che eseguono sempre le mosse migliori, senza commettere errori.

Vengono poste le seguenti domande:

- a. Il giocatore perfetto che incomincia vince per forza?
- b. Il giocatore perfetto che incomincia perde per forza?
- c. Il giocatore perfetto che incomincia vince o perde secondo gli sviluppi casuali del gioco?

Una volta risposto alle domande realizzare il programma che consente di giocare contro il computer.

34. **Calcolo del pi greco con le gocce:** Si espone alla pioggia un foglio di carta quadrato con inscritto un cerchio di diametro pari al lato del foglio. Il valore di pi greco è ricavabile considerando il numero delle gocce cadute nelle due figure (quadrato e cerchio) proporzionale alle aree.

Suggerimento: considerare il cerchio centrato in 0,0 e il quadrato con angoli nei punti 1,1 e -1,-1.

5. Array e stringhe

5.1. Array monodimensionali

| | | | | | | | |
|----|----|----|----|----|----|-----|----|
| 20 | 33 | 50 | 66 | 17 | -3 | 100 | 25 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |

Un generico array

1. Elemento più piccolo in un array di numero reali pieno.
2. Posizione dell'elemento più piccolo in un array di numeri reali parzialmente riempito con indicatore del numero di elementi validi.
3. Posizione dell'elemento più grande in un array di numeri naturali parzialmente riempito con dimensione fornita e elementi non validi che contengono valori negativi.
4. Numero di elementi maggiori della media in un array di numeri interi parzialmente riempito con flag di terminazione che vale -9999.
5. Dato un numero costruisce un vettore con i suoi divisori distinti.
6. Dato un numero costruisce un vettore con i suoi divisori distinti e per ognuno quante volte compare nella scomposizione (ad esempio: $24 = 2 \cdot 2 \cdot 2 \cdot 3$: il 2 tre volte, il 3 una volta).
7. memorizzare in un array di 1000 elementi i primi 1000 multipli di N dato.
8. si richiedono dei valori reali (max 50) fino a che non viene introdotto il valore 0 e poi si mostrano gli stessi valori su due colonne, una con i valori positivi ed una con i valori negativi.

Esempio: se i valori introdotti sono -11.7 1.3 3.05 -5.9 0.4 le due colonne saranno fatte così:

| Positivi | Negativi |
|----------|----------|
| 1.3 | -11.7 |
| 3.05 | -5.9 |
| 0.4 | |

9. si richiedono 24 valori reali e si calcola il numero di volte che capita che un numero è maggiore di tutti i successivi.
10. si chiedono 10 valori reali e si mostrano una riga indicando a fianco quanti valori sono maggiori di esso.
11. memorizzare in un array tutti i numeri inferiori a 10000 che hanno (almeno) tre divisori consecutivi.
12. si richiedono 40 misure di distanze (crescenti) percorse da un'auto dopo 1, 2, 3... 40 secondi. Dopo si richiede ciclicamente, fino a che l'operatore non introduce il valore 0, una distanza e il calcolatore indica quanti secondi sono necessari per percorrerla.

(es. se le misure dopo 21 e 22 secondi sono rispettivamente 390.60 e 426.35 metri e viene richiesta la distanza di 400 metri il calcolatore risponde con il valore di 22 secondi).

13. **La sarta:** Si richiedono cinque misure di lunghezza, che sono le dimensioni di altrettanti scampoli di stoffa, e la lunghezza richiesta per la confezione di un abito. Si fornisce il numero di abiti confezionabili con gli scampoli.



Uno scampolo consente la confezione di tanti abiti quante le volte che la misura dell'abito sta nello scampolo (es. con 8 metri si possono confezionare due abiti che richiedono 3.5 metri e 1 solo che ne richiede 5).

14. **La ditta di confezioni **:** Si richiedono cinque misure di lunghezza, che sono le dimensioni di altrettanti scampoli di stoffa, e altre due lunghezze richieste per la confezione di due abiti. Si fornisce il numero massimo di abiti (che possono essere di una sola delle due misure o in parte di una e parte dell'altra misura) confezionabili con gli scampoli ma **MINIMIZZANDO** gli scarti. Uno scampolo consente la confezione di tanti abiti quante le volte che la misura dell'abito sta nello scampolo (es. con 8 metri si possono confezionare due abiti che richiedono 3.5 metri e 1 solo che ne richiede 5).



15. si acquisiscono due serie di 10 numeri reali che rappresentano una le lunghezze e l'altra le larghezze di 10 pezzi di lamiera. Il programma deve stampare il numero delle lamiere (compreso tra 0 e 9) che hanno area maggiore di un valore dato.

5.1.1. Algoritmi di ordinamento

1. **Ordinamento su altro vettore:** realizzare una funzione che dati due vettori di numeri interi e la loro dimensione riempie il secondo con i valori contenuti nel primo messi in ordine crescente.
Versione 1: senza duplicati.
Versione 2: con duplicati possibili.
2. **Ordinamento con Bubble sort:** realizzare una funzione che dato un vettore di numeri interi e la sua dimensione lo ordina con l'algoritmo del bubble sort.

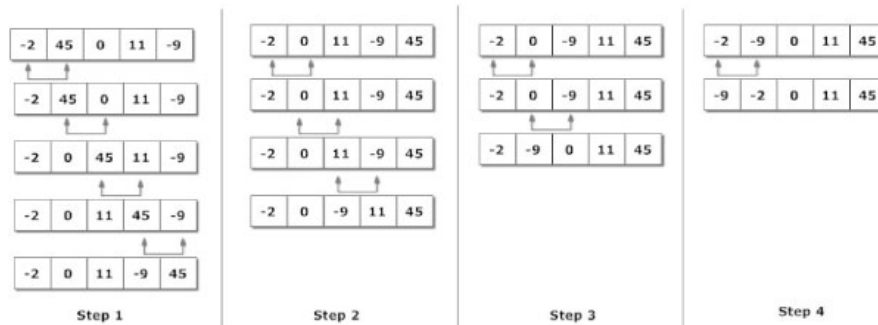


Figure: Working of Bubble sort algorithm

3. **Ordinamento con insertion sort:** realizzare una funzione che dato un vettore di numeri interi e la sua dimensione lo ordina con l'algoritmo dell'insertion sort.

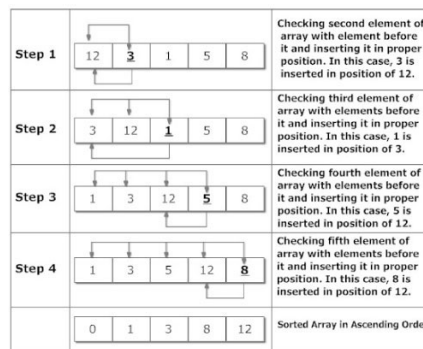


Figure: Sorting Array in Ascending Order Using Insertion Sort Algorithm

4. **Ordinamento con selection sort:** realizzare una funzione che dato un vettore di numeri interi e la sua dimensione lo ordina con l'algoritmo del selection sort.

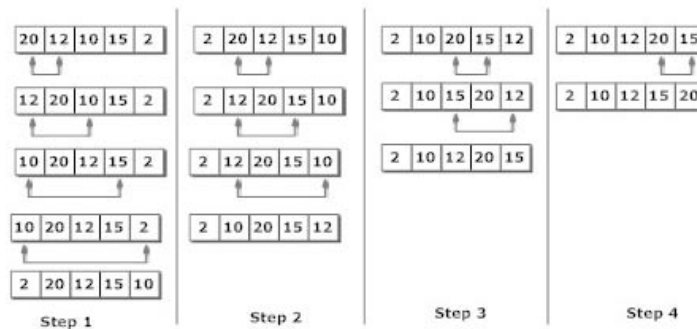
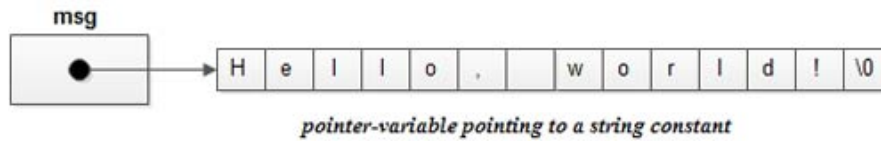


Figure: Selection Sort

5. **Ordinamento con quick sort **:** realizzare una funzione che dato un vettore di numeri interi e la sua dimensione lo ordina con l'algoritmo RICORSIVO del quick sort.
6. **Ordinamento con merge sort **:** realizzare una funzione che dato un vettore di numeri interi e la sua dimensione lo ordina con l'algoritmo del merge sort.

5.2. Stringhe



1. scrivere una funzione che calcoli la lunghezza di una stringa
2. scrivere una funzione che dica quante lettere 'a' ci sono in una stringa
3. scrivere una funzione che dica quante parole ci sono in una stringa
4. scrivere una funzione che copi una stringa su un'altra
5. conta il numero di doppie presenti in una stringa
6. numero di vocali presenti in una stringa
7. Data una stringa di lunghezza massima 80 caratteri stampare il numero di lettere comprese tra 'A' e 'G' che contiene. Supporre i caratteri tutti maiuscoli.
8. sostituire tutte le vocali di una stringa con la lettera 'a'
9. comporre una stringa s3 prendendo alternativamente i caratteri dalle stringhe s1 e s2.
10. invertire il contenuto di una stringa ("Ciao" diventa "oaiC").
11. date due stringhe comunicare se una è l'anagramma dell'altra
12. data una parola comunicare se è palindroma ("anna" e' palindroma)
13. data una frase comunicare se è palindroma (Sono palindrome, ad esempio: "I topi non avevano nipoti", "E nonno dai! Ci porti ai tropici a donnone?", "E' la' Roma, attici da Mafia. Hai fama di citta' amorale", "Era d'uso sudare", "Ai lati d'Isabella e Alle basi d'Italia").
Nella figura la versione ricorsiva dell'algoritmo. Provare a realizzarla.

Palindromi in versione ricorsiva

- Un palindromo è tale se:
- la parola è di lunghezza 0 o 1; **Caso base**
– oppure
- il primo e l'ultimo carattere della parola sono uguali e inoltre la sotto-parola che si ottiene ignorando i caratteri estremi è a sua volta un palindromo **Passo induttivo**
- **Il passo induttivo riduce la dimensione del problema!**

14. **Conta frequenze:** Realizzare un programma organizzato a menù che, date più stringhe di max DIM caratteri ognuna, presenta le seguenti voci:

a) inserire nuovo input.

```
b) frequenze di ogni lettera dell'alfabeto fino a questo momento inserita,
su cinque colonne.
c) lettera o lettere con più occorrenze.
d) lettere presenti in ordine di frequenza.
e) mostra ultima stringa inserita.
f) frequenze relative di ogni lettera dell'alfabeto.
g) numero di frasi fino a questo momento inserite.
x) fine programma.
```

15. verificare se una stringa s1 contiene la stringa s2. restituisce la posizione oppure -1 se non è presente.
16. sostituire nella stringa s1 la stringa s2 con la stringa s3 (se presente). Restituisce: 1 se la sostituzione è avvenuta; -1 se s2 non è stata trovata in s1; -2 se s2 e s3 hanno lunghezza diversa (ed s1 non è stata modificata).
17. date due stringhe produrne una terza prendendo una parola dalla prima e una dalla seconda alternativamente.
18. sostituire nella stringa s1 la stringa s2 con la stringa s3 (se presente). Restituisce: 1 se la sostituzione è avvenuta; -1 se s2 non è stata trovata in s1. Attenzione: s2 e s3 possono avere lunghezze diverse.
19. **Checksum:** Il codice internazionale per i libri (International Standard Book Number, ISBN) è una stringa di 10 cifre unica per ogni libro. L'ultima cifra è determinata a partire dalle altre nove in modo univoco dalla regola che la formula

$$d1 + 2 * d2 + 3 * d3 + \dots + 10 * d10$$

deve essere un multiplo di 11 (d1 indica la cifra più a destra).

Il valore della checksum deve essere un numero tra 0 e 10. La convenzione per ISBN è di indicare il 10 con il valore X (dieci in romano).

Esempio: la cifra di checksum per il codice 020121452 è 5 dato che 5 è il solo valore per cui

$$10 * 0 + 9 * 2 + 8 * 0 + 7 * 1 + 6 * 3 + 5 * 1 + 4 * 4 + 3 * 5 + 2 * 2 + 1 * N$$

è un multiplo di 11.

Realizzare due funzioni: una che verifica se un codice ISBN è corretto e una che date le prime 9 lettere di un codice ISBN calcola la decima e la aggiunge in fondo.

5.3. Array multidimensionali

Altre idee: <http://impararegiocando.com/matematica>.

1. Realizzare un programma in grado di gestire una tabella che rappresenta le temperature di venti località in tutti i giorni di un mese. Il valore -999 indica un valore assente da non considerare.

Deve consentire a) di inserire i valori; b) di inserire valori casuali; c) di svuotare tutti i valori d) di stampare il contenuto; e) di indicare il valore massimo, il valore minimo e la media dei valori presenti; f) di trovare la località con la temperatura media più elevata; g) di mostrare il giorno mediamente più caldo; h) di cancellare tutti i valori relativi ad una località; x) uscire dal programma.

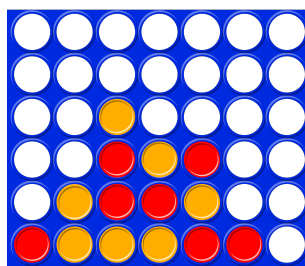
2. **Tris o filetto:** Realizzare il gioco del tris in cui uno dei giocatori è il calcolatore.



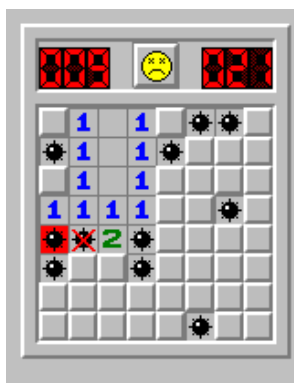
Versione 1: il calcolatore gioca a caso.

Versione 2: il calcolatore cerca di vincere. Se gioca per primo non perde.

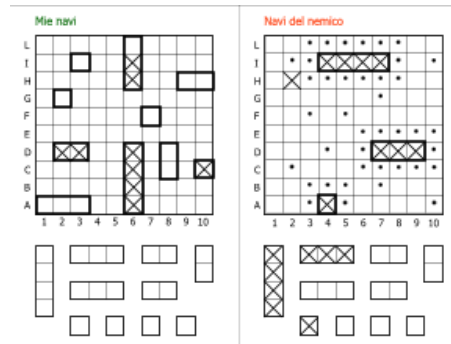
3. **Forza 4:** I giocatori decidono tramite conta chi dei due inizierà il gioco. A turno faranno quindi scivolare la propria pedina nei binari prestabiliti dalla scacchiera cercando di allineare quattro pedine in fila orizzontale, verticale o obliqua. Nel frattempo devono stare attenti alle mosse dell'avversario per evitare che anch'esso riesca a fare quattro impilando le proprie. Vince la partita chi allinea per primo in orizzontale, verticale o obliquo quattro pedine del proprio colore. Nel caso nessuno dei due partecipanti al gioco riuscisse a fare #quattro# la partita finisce in parità e se ne comincia un'altra.



4. **Campo minato:** Realizzare il solitario del campo minato (ora prato fiorito...) in forma testuale.



5. **Battaglia navale:** Realizzare il gioco della battaglia navale.



Versione 2: uno dei giocatori è il calcolatore.

6. **Gioco del 15:** consiste nel cercare di riordinare in sequenza crescente le tessere numerate da 1 a 15 in una tabellina di forma quadrata di 4 righe e 4 colonne. Può essere spostata nella casella vuota una tra le tessere ad essa vicine.



6. Strutture Dati

6.1. Struct e typedef

1. definire una struttura per memorizzare un orario (ore, minuti, secondi) e scrivere le funzioni necessarie per assegnare un valore, aggiungere un intervallo di tempo, restituire la differenza, in secondi, tra due orari.
2. realizzare una struttura dati in grado di memorizzare l'orario delle lezioni di una classe (giorno, ora, materia, docenti).

6.2. Allocazione dinamica

1. Scrivere una funzione che a seconda di una lettera che identifica il tipo (i=int, f= float, c=char, s=stringa) e un numero che rappresenta la dimensione, crea un vettore del tipo scelto e della dimensione proposta e lo riempie di altrettanti valori casuali (per le stringhe mette parole con un numero casuale di lettere alfabetiche casuali compreso tra 5 e 10).
2. Usando l'aritmetica dei puntatori per scorrere il vettore, dati un vettore di float e la sua dimensione cambiare ogni elemento negativo con il valore 0. Restituisce il numero di elementi cambiati.
3. Scrivere una funzione che crea un vettore di numeri float della dimensione passata in input e inserisce in esso i prii valori interi a partire da 1 e incrementandoli di 4 ogni volta.
4. Scrivere una funzione

```
char * trovaESostituisci(const char *s1, const char * s2, const *s3)
```

che cerca in una stringa una sottostringa e costruisce una nuova stringa in cui ogni occorrenza di s2 in s1 è sostituita con s3. Se si volesse cambiare il valore direttamente in s1 come si potrebbe procedere? Fare una analisi dettagliata.

- **Costruttore:** Data una struct Ora, realizzare una funzione costruisciOra che, forniti i valori di ore, minuti e secondi crea una struct Ora e ne restituisce l'indirizzo dopo aver inserito in essa i valori passati.

Realizzare una seconda funzione costruisceOraVuota restituisce una struct Ora con valori di ora, minuti e secondi pari a zero.

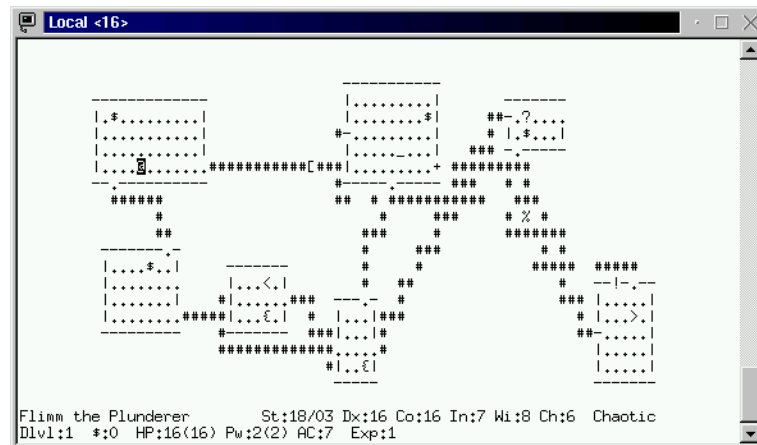
6.3. Strutture dati complesse

1. **Lista:** Realizzare una lista in grado di contenere parole di lunghezza qualunque. Prevedere funzioni per aggiungere, cercare, togliere parole e visualizzare il contenuto della lista.
2. **Albero binario:** Realizzare un programma in grado di realizzare un albero binario in grado di memorizzare dei numeri interi. Non possono essere presenti duplicati. Realizzare le funzioni necessarie a: a) inserirvi 100 numeri casuali tra -100 e 100; b) stampare il contenuto in ordine crescente (pre-order); c) stampare il contenuto in ordine decrescente (post-order); d) cercare se è presente un valore dato; e) inserire un valore dato; f) eliminare un valore.

3. Discutere le problematiche della possibilità di inserire duplicati nell'albero dell'esercizio precedente e provare a modificarne una copia per consentire questa possibilità.

6.4. Esercizi di riepilogo

1. **Gioco di ruolo:** Realizzare un gioco di ruolo in cui i diversi personaggi si muovono in un labirinto (cercare rogue o nethack su internet per degli esempi).



Una schermata di NetHack

2. **Briscola:** Realizzare un programma che consenta di giocare a briscola con il calcolatore.

7. Files

7.1. Esercizi di base

Argomenti:

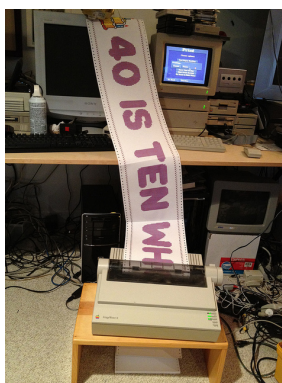
- **File binari:** : funzioni di utilità, ricerca, aggiunta, cancellazione di valori.
 - **File di testo:** : confronto con i file binari per la memorizzazione di valori numerici; frequenza delle lettere e delle parole.
1. **Dimensione di un file:** Realizzare una funzione che dato il nome di un file restituisce la dimensione in byte.
 2. **File binari:** Realizzare un programma che consente due operazioni: a) scrive su un file binario "interi.dat" una sequenza di numeri interi (ad esempio i numeri da 1 a 100, oppure 100 numeri casuali); b) legge lo stesso file visualizzando sul monitor i numeri letti, uno per riga, e al termine il numero di elementi letti. Se il file non è stato riempito l'operazione di lettura indica "file vuoto".
 3. **File binari con numeri reali:** Duplicare l'esercizio precedente utilizzando numeri reali anziché interi. La dimensione del file cambia? Perché?
 4. **Ricerca valore:** Realizzare una funzione che cerca un valore in un file contenente un elenco di numeri reali. La funzione restituisce la posizione del valore (0 se nella prima posizione, 1 se nella seconda e così via), oppure -1 se il valore non è presente nel file.
 5. **Aggiunta di un valore:** Realizzare una funzione che cerca un valore in un file contenente un elenco di numeri reali e, se non viene trovato, lo aggiunge in fondo al file. La funzione restituisce 0 se il valore è stato aggiunto o -1 altrimenti.
 6. **Cancellazione di valore:** Realizzare una funzione che cerca un valore in un file contenente un elenco di numeri reali e, una volta trovato, lo elimina dal file. La funzione restituisce 0 se il valore è stato tolto o -1 altrimenti. SUGGERIMENTO: forse occorre riscrivere completamente il file.
 7. **File di testo:** Realizzare un programma che consente due operazioni: a) scrive su un file di testo "interi.txt" un elenco di valori interi (casuali, oppure letti da input), separati tra loro da uno spazio; b) legge lo stesso file visualizzando sul monitor una tabella di due colonne, con intestazione "pari" e "dispari", contenente i numeri letti, uno per riga, nella colonna corretta. Sotto la tabella indica il numero di elementi letti.
 8. **File di testo e file binari:** Confrontare le dimensioni tra un file binario contenente un dato numero di valori interi e un file di testo in cui gli stessi numeri sono memorizzati separati da uno spazio. Quale dei due è più grande? Cambiando i valori memorizzati le dimensioni dei file cambiano a loro volta?
 9. **Parole in un file:** Realizzare una funzione che dato il nome di un file di testo conta il numero di parole presenti nel file.
 10. **Top Ten:** Molti giochi on line consentono di indicare il proprio nome e al termine di ogni partita visualizzano i 10 migliori punteggi ottenuti con vicino il nome del giocatore. La "top ten" è solitamente memorizzata in un file. Fare una analisi approfondita del problema e

presentare due soluzioni: una che prevede l'uso di file binari e una che prevede l'uso di un file di testo. Realizzarle entrambe e discutere sui pregi e i difetti di ognuna di esse.

11. **Frequenze delle lettere:** Scrivere un programma che riceve in input un file di testo e conta le occorrenze di ognuna delle lettere dell'alfabeto. Alla fine mostra due tabelle con tre colonne: lettera, frequenza assoluta e frequenza relativa (numero di occorrenze sul totale delle lettere). Le due tabelle sono in ordine alfabetico la prima e in ordine di frequenza la seconda.
12. **Conta occorrenze:** Scrivere un programma che riceve in input un file di testo e costruisce un secondo file che contiene tutte le parole presenti nel primo con vicino il numero di occorrenze di ognuna di esse.

7.2. Esercizi complessi

1. **Banner:** Negli anni '80 era frequente realizzare dei banner cartacei con la carta a modulo continuo delle stampanti ad aghi. Per fare ciò, scelta la frase da stampare, ogni lettera veniva stampata ruotata e molto in grande utilizzando più volte la stessa lettera.



Realizzare un programma che consente all'utente di inserire il testo e scrive su un file il banner, supponendo siano disponibili 80 caratteri per ogni riga (cosa comune a quei tempi).

```

AAA   SSSSS  CCCCC  IIIII  IIIII  TTTTTTT  tt
AAAAA  SS    CC    C   III   III   TTT    eee  xx  xx  tt
AA  AA  SSSSS  CC    III   III   TTT   ee  e   xx  tttt
AAAAAAA      SS CC    C   III   III   TTT  eeeee  xx  tt
AA  AA  SSSSS  CCCCC  IIIII  IIIII  TTT   eeeee xx  xx  tttt

```

2. **Generatore di parole con i numeri telefonici:** Le tastiere telefoniche standard dei telefoni cellulari contengono le cifre da zero a nove. A ognuna delle cifre da 2 a 9 sono associate tre o quattro lettere (vedi tabella).

| | | | |
|---|-------|---|---------|
| 2 | A B C | 6 | M N O |
| 3 | D E F | 7 | P Q R S |
| 4 | G H I | 8 | T U V |
| 5 | J K L | 9 | W X Y Z |

Per molte persone la memorizzazione dei numeri di telefono è una operazione difficile, per cui spesso utilizzano questa corrispondenza per sviluppare parole di sette caratteri che corrispondano ai propri numeri telefonici (ciò vale negli Stati Uniti, ma può essere adattato anche da noi). Per esempio, una persona con numero di telefono 468-6734 potrebbe usare la tabella e individuare la parola "INUMERI".

Le aziende cercano per questo di avere numeri di telefono, dalle ditte di telefonia, che possano essere facilmente ricordati dai propri clienti. Scrivere un programma che, dato un numero di telefono a sette cifre, scriva su un file tutte le possibili combinazioni di parole ottenibili con quel numero. Al massimo sono 16384 (perché?).

Evitare i numeri di telefono con le cifre zero e uno.

3. Usando un dizionario completo, modificare l'esercizio precedente in modo che mostri solo le parole presenti nel dizionario.
4. **Analizzatore di file:** Scrivere un programma che estragga da un file HTML tutti i tag e li mostri sull'output. Un tag inizia con '<' e termina con '>'.