**QUAL’ È LA DIFFERENZA TRA UN FILE BINARIO E UN FILE DI TESTO?**

Entrambi memorizzano i dati come una serie di bit, l a differenza sta nel fatto che i due tipi di File ***codificano i dati in modo diverso***:   
i bit del ***File di testo*** cont solo dati testuali , rappresentati da dei caratteri Ascii e in esso il numero da inserire fa variare la quantità di memoria (0 occupa 1 byte mentre 92134 occupa cinque byte ) mentre quelli del   
***File binario***  contiene anche immag, video, testi,(sia dati testuali che in codice binario) rappresentati in base all’applicazione come, ad esempio, Word o Block note, inoltre i dati sono memorizzati nello stesso modo in cui

si trovano in memoria, per cui, per esempio, un intero occupa su un file binario sempre sizeof(int) byte, indipendentemente dal suo valore .

Lo spazio occupato è indipendente dal valore che si vuole rappresentare.

**PERCHE’ È MEGLIO USARE UN OUTPUT.BIN?**

Il file binario **viene memorizzato in memoria con una sequenza di byte.  
Sono leggibili a multipiattaforma**Se vogliamo memorizzare il numero 1365, in memoria troveremo 2 byte.

I vantaggi dei file binari rispetto a quelli di testo sono:

**minore dimensione (in media),**

**facilità di modificare il file,**

**facilità di riposizionarsi nel file.**

//Gli svantaggi sono la non portabilità da un tipo di calcolatore a un altro, e il fatto che non si può creare o modificare un file binario usando un editor di testi.

//Perché gli output.bin sono leggibili a multipiattaforma in quanto tutte usano il linguaggio binario.//

**CHE COS’E’ UN PROGRAMMA?**

Un programma è una serie di istruzioni che possono essere eseguite da un computer. La costruzione di un programma è usualmente associata alla necessità di risolvere in maniera efficiente uno o più problemi, mentre il **processo è** un **programma** in esecuzione su un determinato elaboratore.

//È la codifica di un algoritmo in un determinato linguaggio macchina e il risultato è un software ovvero un programma eseguibile.//

**COME SEPARARE LE PAROLE IN UNA STRINGA?**

Con la funzione Strtok.

La funzione strtok fa parte della libreria standard C definita nel file di intestazione <string.h>.

Spezza la stringa data in token divisi dal delimitatore specificato.

strtok accetta due argomenti: un ***puntatore alla stringa da tokenizzare*** come primo parametro e la ***stringa delimitatore come secondo.***

**COSA FA UN CONTATORE?**

È una variabile che noi utilizziamo quando usiamo un ciclo for, conta , incrementa se si verifica una condizione richiesta.

**QUANDO SI USANO I PUNTATORI?**

È necessario utilizzare i puntatori in qualsiasi **punto in cui è necessario ottenere e passare l'indirizzo in un punto specifico della memoria**. È inoltre possibile utilizzare i puntatori per navigare negli array:   
Un array è un blocco di memoria contigua che è stato allocato con un tipo specifico.

I puntatori sono utili quando si scrivono le funzioni. Le funzioni prendono i loro argomenti ‘in base al valore’, il che significa che ottengono una copia di ciò che viene passato e se una funzione assegna un nuovo valore a uno dei suoi argomenti che non influirà sul chiamante.

(esempio: passare un array o una cella di un array come argomento a una funzione)

**COS’E’ IL LINKER?**

È un software che collega alle istruzioni del nostro programma, il codice delle librerie complesse richiamate all’interno del file sorgente.

**QUALI SONO LE DIRETTIVE DI COMPILAZIONE?**

Sono **#include** (serve al compilatore che sta per usare delle librerie) e   
**#define** (dice al compilatore il valore da sostituire ad una variabile ogni volta che la si incontra).

**COME SI SALVA UNA MATRICE CON FWRITE?**

**Includere libreria stdio.h**

Fwrite(buf cioè matrice in cui scrivere, sizeof(int) sarebbe dimensione, len sarebbe elementi, fp);

**COME SI ALLOCA DINAMICAMENTE UNA MATRICE?**

Con la funzione malloc 1 paramentr=numero di byte da allocare che poss passare come sizeof(tipo).  
**Includere prima libreria stdlib.h**

**QUALI SONO LE CARATTERISTICHE DEL LINGUAGGIO C?**

Imperativo ad alto livello, strutturato(ci sono dei costrutti, while for ), case sensitive, elementare, portabile, standard, tipizzato.

**COS’E’ UN ALGORITMO?**

È una sequenza di passi non ambigua che portano alla realizzazione di un determinato compito.

**QUALI SONO LE DIMENSIONI DEI TIPI?**

Char caratteri Ascii (1 byte)

Int interi (4 byte)

Float reali (4 byte)

Long (4 byte)

Double reali a doppia precisione (8 byte)

**DI CHE TIPO POSSONO ESSERE I COSTRUTTI?**

**P**ermettono di combinare tra loro istruzioni elementari creando cosi istruzioni complesse o blocchi di istruzioni, controllando il flusso della loro esecuzione. I costrutti di controllo fondamentali si possono classificare in:

* [Esecuzione in sequenza](http://www.dacrema.com/Informatica/Costrutti_controllo.htm)
* [Costrutti iterativi](http://www.dacrema.com/Informatica/Costrutti_controllo.htm) :comando che dice al processore di ripetere un’istruzione un determinato numero di volte(while, do while, while do)
* [Costrutti di selezione](http://www.dacrema.com/Informatica/Costrutti_controllo.htm) :insieme di condizioni e istruzioni condizionate (costrutto if)

**OPERATORI LOGICI**

Sono operatori logici && (AND) e || (OR)

**TEOREMA DI DE MORGAN**

Il negato di un’espressione booleana si ottiene negando tutti i termini o sostituendo ogni operatore logico con il proprio duale AND-OR e viceversa.

**COSA SIGNIFICA OTTIMIZZARE?**

Significa ridurre al minimo il numero di istruzioni al fine di ottenere il medesimo risultato

**DIFFERENZA TRA COMPILAZIONE E LINKING**

Partendo dal ciclo di sviluppo di un software, dopo aver scritto un programma c’è una fase chiamata compilazione e svolta dal compilatore che traduce il file sorgente nel linguaggio macchina scelto e quindi in un nuovo file chiamato oggetto.   
Tuttavia, questo File potrebbe contenere informazioni più complesse rispetto a quelle contenuto nel processore. Quindi il linker(software) collega il codice delle istruzioni scritte da noi con quello contenuto in apposite librerie.

**COS’E’ L’OPERATORE UNARIO DI DEREFERENZAZIONE?**

Si applica ad una variabile di tipo puntatore, cioè quelle variabili che contengono un indirizzo e restituisce il contenuto dell’oggetto puntato. Non c’entra niente con l’operatore \* di moltiplicazione. Restituisce l’indirizzo di memoria di una variabile.

**COSA FA LA FUNZIONE STRSTR?**

Strstr è una funzione della libreria string.h, che divide la stringa in sottostringhe.

Char \*strstr (char \*str1, const \*str2); Due paramentri stringa originale e sottostringa da trovare.

Cerca la sottostringa str2 in str1 e ritorna il puntatore alla posizione in str1 o NULL se la sottostringa non è presente.

**COME SALVARE UN ELEMENTO DI UNA STRUCT (SIA IN UN BIN CHE IN UN TXT)?**

In **binario con fwrite** (nome struct, size of(Mystruct) , elementi , fp)

mentre **in txt con fprintf** (fp , tipo(cioè %d, %d) ,argomenti).

**DIAGRAMMI DI FLUSSO:**

sono un linguaggio grafico visuale e non simbolico per la rappresentazione di un algoritmo. È composto da ***blocchi*** che rappresentano i passi dell’algoritmo. Le ***frecce*** sono **monodirezionali** da una sorgente ad una destinazione.

**QUALI SONO LE OPERAZIONI ELEMENTARI**

Sommatoria dei primi N numeri naturali, fattoriale, conversione da decimale a binario e “zero” di una funzione.

**COME SI TRASORMA UN CHAR IN INT E VICEVERSA? (fai cast implicito)**

Attraverso il cast esplicito per fare da char a int perché char è < di int e per farlo basta mettere tra parentesi (int) quindi:

char c=’d’;

int a=(int) c;

Invece da int a char va bene anche quello implicito poiche int>char:

int n=65;

char c=n;

**DI CHE TIPO È UN VETTORE?**

È un tipo di dato strutturato costituito da un insieme omogeneo di dati tutti dello stesso tipo.

*Il tipo definisce quanti byte servono per memorizzare ogni elemento.* Nel caso di un vettore di interi, lo spazio di memoria necessario è sizeof(int)\*N dove N è la lunghezza del vettore. Il compilatore deve conoscere la dimensione massima per riservare spazio in memoria.

**COSA SONO LE VARIABILI GLOBALI?**

Sono le variabili usate fuori dal main e che vengono viste da tutti i sottoprogrammi, può essere ad esempio cambiato e quindi sbaglierà valore di uscita.

**COME SI SALVA UN ARRAY DI STRUCT IN UN FILE BINARIO?**

Fwrite(vs nome struct , sizeof(mystruct) dimensione , len, fp);

**COS’E’ LA RICORSIONE?**

Una funzione si dice ricorsiva se chiama a sé stessa direttamente o indirettamente.  
Esiste ricorsione lineare,in coda o multipla.   
APPROCCIO ALLA PROGRAM.RICORSIVA:Dividi,Impera,Combina.

È un’alternativa alle strutture iterative come for o while e un esempio in cui conviene usarla è il calcolo del fattoriale.

**COME VIENE DEFINITO UN ALGORITMO RICORSIVO?**

Dividi,Impera,Combina.  
Prima si definisce come risolvere dei problemi analoghi a quello di partenza ma che hanno dimensione più piccola e possono essere risolti in modo più semplice “casi base”. Si definisce poi come ottenere la soluzione del problema di partenza combinando la soluzione di uno o più casi base

**PRINCIPIO DI INDUZIONE**

Sia P una proprietà, se p(k) è vera per k=1 (base dell’induzione) ed è vera per un valore generico n (ipotesi induttiva), se riusciamo a dimostrare che p(n+1) è vera allora p(k) sarà vera per qualsiasi valore di k.

**QUAL’E’ L’APPROCCIO ALLA PROGETTAZIONE RICORSIVA?**

Divide et impera

**CHE COS’E’ IL DEBUG?**

È la revisione del codice per trovare errori eseguendolo passo per passo e vedendo tutti i passaggi che compie.

**COS’E’ IL CAST?**

È un’operazione che forza il cambio di tipo.

**CHE DIFFERENZA C’E’ TRA CAST IMPLICITO ED ESPLICITO?**

* Implicito: viene compiuto automaticamente da C dalla variabile più piccola a quella più grande nell’ordine DOUBLE(8 byte) > FLOAT(4 byte) > LONG(4 byte) > INT(4 byte) > SHORT(2 byte) > CHAR (1 byte)
* Esplicito: viene compiuto dal programmatore e, sebbene abbia la precedenza rispetto al cast implicito, comporta una diminuzione di informazione e quindi di precisione.

Una conversione di implicita è quindi la conversione da char ad int mentre esplicita è da int a char.

**COS’E’ IL COSTRUTTO SWITCH-CASE?**

È un costrutto che consente di implementare decisioni multiple e si basa sul confronto fra il risultato di un’espressione (int char ecc) e dei valori costanti.

**COSA FANNO LE FUNZIONI MALLOC E CALLOC?**

Per usare la memoria heap usiamo la funzione malloc che ha come parametro una dimensione che alloca in memoria e ritorna un puntatore a void, corrispondente al punto di inizio in memoria della porzione riservata alla dimensione allocata. Se la dimensione passata non può essere allocata ritornerà un puntatore nullo.

La memoria allocata così è ***persistente*** cioè continuerà ad esistere fino alla fine del programma o fino a quando non sarà esplicitamente deallocata dal programmatore con la funzione free.

La calloc invece, pur essendo molto simile, chiede due parametri: il numero di elementi e la dimensione del loro tipo. Essa inizializza poi a 0 ogni byte del blocco di memoria occupato.

**ALLOCAZIONE DINAMICA DI MEMORIA:**

Il C permette di lavorare sugli spazi di memoria attraverso i puntatori. Esistono due tipi di memoria:

* Stack: memoria statica che contiene tutto ciò che è stato dichiarato dal codice
* Heap: è una parte dinamica in cui la dimensione degli elementi può cambiare a Runtime

La prima è più veloce in quanto gli elementi sono già stati allocati in memoria.

**A COSA SI DEVE STARE ATTENTI QUANDO SI USA UNA FUNZIONE RICORSIVA?**Ogni volta che la funzione viene richiamata si tratta si della stessa funzione dal punto di vista logica ma non per quanto riguarda la memoria. Non conviene dunque utilizzare per ogni problema un approccio ricorsivo poiché potrebbe occupare velocemente buona parte della memoria se non tutta. Potrebbe inoltre capitare che, a causa di un errore in fase di codifica, la chiamata ricorsiva del sottoprogramma non termini mai e dunque in memoria vengano creati infiniti sottoprogrammi uguali che satureranno e ci obbligheranno a riavviare il computer.

**FALLIMENTO ALLOCAZIONE**

È importante verificare sempre che l’allocazione abbia avuto successo e che non abbia restituito NULL.

In caso contrario si rischia di usare il puntatore NULL come se puntasse a memoria allocata e si provocherebbe errori.

L’ allocazione della memoria non riesce quando essa è piena.

**COSA SONO FPRINTF E FSCANF?**

Sono gli equivalenti di printf e scanf per lettura/stampa su file e vogliono un parametro in più che sarebbe il puntatore a file prima degli apici.   
Fprintf trasforma inoltre le variabili in stringhe e quindi in caratteri ASCII.

**COS’E’ UNA STRUCT?**

Le strutture come gli array sono un tipo di dato derivato cioè costituito da altri tipi di dati. Tuttavia, mentre un array è costituito da elementi omogenei, la struct è invece costituita da elementi eterogenei. La sintassi è struct nomestruct {tipomembro nomemembro 1,2,3}. Gli elementi della struct sono chiamati campi riconosciuti / indicati da nomi .

**COS’E’ TYPEDEF?**

È una keyword che ci permette di rinominare un qualsiasi tipo creandone quindi un sinonimo.

Questo può essere molto utile per rendere più leggibile un programma soprattutto quando possiede espressioni logiche complesse.

**COS’E’ ENUM?**

È una keyword che serve a introdurre un nuovo tipo, come struct, ma orientato specificamente a dichiarare variabili i cui valori possibili sono ristretti a un insieme finito e a priori noto.   
Si considerino i seguenti esempi:  
**enum** settimana {lunedi, martedi, mercoledi, giovedi, venerdi, sabato, domenica};

**COME VENGONO PASSATE LE MATRICI IN UNA FUNZIONE?**

*Importante: nella funzione va sempre specificata la seconda dimensione (statica) della matrice.   
Se non è specificata non compila.*  
Quando un array monodimensionale viene passato come argomento di una funzione, viene passato solo il puntatore al suo primo elemento in quanto questo puntatore e il suo tipo sono necessari a calcolare l’indirizzo di un qualsiasi elemento dell’array.   
Una matrice è invece un array di array. Si può quindi considerare come un array i cui elementi sono le righe di cui però è necessario conoscere la lunghezza.  
Delle *righe si può non specificare la dimensione* mentre delle colonne si perché ogni tot byte il compilatore passa alla riga successiva quindi ***deve sapere il numero preciso di byte delle colonne*** perché potrebbe altrimenti sfondare il programma.

**CHE DIFFERENZA C’E’ TRA STRUCT E UNION?**

Nella sintassi non c’è differenza, tuttavia, nella struct posso usare tutti i campi contemporaneamente mentre nella Union posso usarne solo uno per volta. I campi infatti coincidono hanno cioè lo stesso indirizzo di memoria. Questo implica che l’occupazione di memoria di una union coincide con quella del membro di dimensione maggiore. Ogni volta che assegniamo un valore a un campo, lo sovrapponiamo a quello precedente.

**QUALI SONO I PARAMETRI DI UNA FUNZIONE?**

I parametri dichiarati nella definizione sono detti **parametri formali e in essi** devono essere inclusi i tipi di dati dei valori di ricezione . Possono essere d’ingresso, uscita, ingresso-uscita. Il tipo di ritorno può essere un void se non torna un risultato oppure può essere un tipo primitivo che quindi torna un risultato.

Quando però viene chiamata i parametri si dicono effettivi e in essi non vi è alcuna menzione dei tipi di dati. Viene menzionato solo il valore.

**COME SI ORGANIZZANO LE LIBRERIE?**

* File.h è un semplice [file di testo](https://it.wikipedia.org/wiki/File_di_testo) che contiene i [prototipi](https://it.wikipedia.org/w/index.php?title=Prototipo_di_funzione&action=edit&redlink=1) delle [funzioni](https://it.wikipedia.org/wiki/Funzione_(informatica)) definite nel relativo file.c.
* File.c contiene le definizioni delle funzioni
* File.o si ricava compilando il file.c ed è quindi la libreria compilata

La chimaerai nel tuo file con le virgolette così : #include “mialibreria.h”;

**QUALI LIBRERIE CONOSCI?**

1.Stdio.h (per operazioni di input e output) ,

2.Stdlib.h (unità generale, per malloc, conversione dei tipi),

3.string.h (gestione di stringhe),

4.time.h(gestione della data e dell’ora),

5.math.h(funzioni matematiche).

**APERTURA DI UN FILE, PARAMETRI E MODALITA’**

Le operazioni effettuabili su un file sono apertura,chiusura,scrittura e lettura.

Per quanto riguarda ques’ultima si usa la funzione fopen, fornita da stdio.h, che vuole due parametri, il nome del file e la modalità ovvero l’operazione che si vuole svolgere su un file e che può essere r, r+, w, w+, a, a+.

**IN CHE MOMENTO FOPEN PUO’ RITORNARE NULL?**

Qualora il programma non riesca a trovare il file o qualora non avessimo i permessi per aprirlo.

**CHE PARAMETRI VUOLE FWRITE?**

Indirizzo del dato da scrivere , dimensione, elementi, fp.

Buffer = area di memoria temporanea .

È il metodo di scrittura su file binario che prende un array di byte e lo invia al file attraverso canale creato con fopen e chiuso con fclose.

**IN QUALI FILE SI USANO FTELL E FSEEK?**

Nei File binari:

* Fseek consente di muovere il puntatore di letture e/o scrittura in qualunque posizione e vuole 3 parametri: fp, n, mode(0 inizio,1 attuale.2 fine)
* Ftell ritorna la posizione corrente del file pointer espressa come numero di byte, restituisce un valore negativo se c’è un errore. ***Ftell(fp);***

**CHE COS’E’ LA LIBRERIA GTK+?**

Acronimo che sta per **GIMP ToolKit** in [informatica](https://it.wikipedia.org/wiki/Informatica), è un [toolkit](https://it.wikipedia.org/wiki/Toolkit) (insieme di strumenti, di cui il principale è la libreria libgtk) per la creazione di [interfacce grafiche](https://it.wikipedia.org/wiki/Interfaccia_grafica). Sviluppato in [C](https://it.wikipedia.org/wiki/C_(linguaggio)). Il motivo principale per cui la usiamo è perché è un open-source funzionale come C e non ad oggetti come Java.

**COMPLESSITA’COMPUTAZIONALE**Si definisce complessità computazionale l’ordine di grandezza della funzione T(n) nel modello RAM .

Il modello RAM è un calcolatore con un solo processore in cui le istruzioni vengono eseguite in stressa sequenza senza parallelismo e necessitano di un tempo unitario.   
T(n) è il numero di istruzioni da eseguire in funzione della dim del problema , si usano le notazioni asint per confrontarli che sono omega, teta e o.

Fanno parte di o se la funzione moltiplicata per una costante c1 è maggiore definitivamente.

fanno parte di teta se la funzione moltiplicata per una costante c1 è maggiore definitivamente di f(n) e moltiplicata per c2 è minore definitivamen.

fanno parte di omega se la funzione moltiplicata per una costante c è minore definitivamente di f(n).

**OPERATORI SU BIT**

Sono spesso usati per impostare, cancellare o capovolgere singoli bit in programmi che controllano l’hardware del computer. Gli operatori bit a bit vengono spesso utilizzati anche nella crittografia dei dati   
Il C mette a disposizione degli operatori che lavorano su numeri di tipo intero manipolando il dato a livello di singolo bit. Non si applicano quindi a float e double o a dati di tipo strutturato. Sono:

* & AND (confronta ogni bit del 1° operando con il bit corrisp del 2°operando viene impostato su 1 se entrambi sono 1 altrimenti 0)
* | OR (se uno vale 0 e l'altro 1 o entrambi 11 allora 1 altrimenti 0)
* ^ OR ESCLUSIVO O XOR(se uno vale 0 e l'altro 1 allora 1 altrimenti 0)
* NOT
* << SCORRIMENTO A SINISTRA
* >> SCORRIMENTO A DESTRA

**DIFFERENZA TRA ARRAY E STRINGA E COME INIZIALIZZARE LE DUE:**

L’array è un insieme di elementi omogenei tutti dello stesso tipo raggruppati sotto un’unica variabile.

la stringa è invece un array di caratteri (tipo char).

L’inizializzazione dell’array può avvenire in due modi:

* Int voti [6] = {8,8,7,6,5,4}
* Int voti [] = {8,8,7,6,5,4} senza specificare la lunghezza che sarà contata automaticamente dal compilatore

L’inizializzazione della stringa pure può esser fatta in due modi:

* Char s1 [20] = “macchina” la dimensione max è 20 mentre il riempimento è 9 perché considera anche \0 chiamato tappo della stringa
* Char s2[] = “macchina” sia il riempimento che la dim max è 9

**DIFFERENZA TRA SCANF, FGETS, GETS**

* ***Scanf*** legge da console una sequenza di caratteri fino all’invio o al primo spazio ed inserisce all’interno della stringa i caratteri + \0;
* ***Gets*** legge tutta la stringa con spazi inclusi, tuttavia, è pericolosa in quanto non possiede una dimensione massima ed elimina l’andare a capo;
* ***Fgets*** ha 3 parametri e legge fino allo \0 (frase, len, stdin).

Stdin è il luogo da cui reperire i caratteri per riempire la frase.

**PERCHE GETS POTREBBE DARE UN WARNING?**

Perché non inseriamo la dimensione del buffer e quindi potrebbe verificarsi che l’utente inserisca una stringa troppo lunga che sfonderebbe il programma

**DIFFERENZA TRA WHILE E FOR**

For si usa quando conosciamo già il numero di volte che il ciclo deve ripetersi.

La differenza dal while è che quindi non abbiamo bisogno di cambiare la condizione all’interno del blocco perché il 3° parametro cambia automaticamente l’istruzione.

**INSERTION SORT**

Partendo dall’indice di posizione 1, considera l’elemento con quello precedente, se il numero è minore viene scambiato mentre quello precedente viene trattenuto al di fuori dell’array in una variabile temporanea per poi essere confrontato con gli elementi precedenti. Gli elementi vengono quindi confrontati uno per volta, inseriti nella variabile temp e sostituiti negli spazi vuoti quando l’elemento con cui vengono confrontati è minore.

**SELECTION SORT**

Ogni volta cerca il minimo dell’array e, il più piccolo lo inserisce nella prima posizione, il secondo minimo nella seconda e così via.

**BUBBLE SORT**

Confronta il numero nella posizione 1 con il precedente, se maggiore lo sostituisce.

Poi confronta le altre coppie di numeri così che, come una bolla, il valore maggiore salga all’ultimo posto.

**COS’E’ UN PUNTATORE?**

Sono tipi di dati, come int, char, float, che rappresentano la posizione attraverso gli indirizzi di memoria, di elementi del programma come variabili, struct e sottoprogrammi.

*Sono quindi variabili abilitate a contenere un indirizzo di memoria di un’altra variabile (puntata) la quale può essere a sua volta un puntatore*. È possibile assegnare ad un puntatore solo due valori: **NULL se non punta a niente e l’indirizzo di una variabile esistente restituita dall’operatore &.**

**COME FUNZIONA L’ARITMETICA DEI PUNTATORI?**

Il puntatore deve essere immaginato come un intero che ci indica l’indirizzo di memoria.  
Le operazioni possibili tra puntatori sono le sole di   
**incremento (***Incremento per andare da un indir. più basso ad un indirizzo più alto***)** e **decremento (**a uno più basso**)**.   
Tuttavia, il valore numerico contenuto dal puntatore non viene mai modificato.

**SCAMBIO DI VARIABILI (TRAMITE PUNTATORE)**

Abbiamo 4 variabili, due variabili di tipo intero a, b e due variabili di tipo puntatore a int p1 e p2.

Utilizzando una variabile ausiliaria e l’operatore unario di referenziazione possiamo effettuare lo scambio.

**COME SI PUO’ CONVERTIRE UNA STRINGA IN UN INT O IN UN DOUBLE?**

Per convertire una stringa in un intero si può utilizzare la funzione atoi.

Invece, per convertire una stringa in un double si può usare la funzione atof.

**CICLO A CONTEGGIO/A INTERRUZIONE**

I cicli iterativi, for(conteggio) e while(interruzione) sono rispettivamente dei cicli a conteggio e ad interruzione perché il for ha un numero determinato di ripetizioni, invece, il while smette di ripetersi non appena una condizione non si verifica più.

**IL FILE.O VIENE CREATO DURANTE LA FASE DI ESECUZIONE O DI COMPILAZIONE?**

Il file.o viene creato durante la compilazione, prima del linking, fase successiva il cui fine è quello di restituire un file eseguibile (.exe), ottenuto grazie al collegamento (linking, appunto) dei file oggetto tra di loro.

**PUNTATORE A PUNTATORE**

Il puntatore è una variabile con un tipo, per cui il suo indirizzo è un puntatore a un puntatore.

Per memorizzare per esempio l'indirizzo di un puntatore a intero in una variabile, questa deve essere di tipo puntatore a puntatore a intero. Questo tipo si scrive int \*\*.

**Cos'è Quicksort**

Quicksort è un algoritmo interno che utilizza la "tecnica divide et impera".   
È anche chiamato **una sorta di scambio di partizioni**. Usa un elemento chiave chiamato pivot per confrontare e partizionare gli elementi dell'array. Gli elementi con un valore inferiore rispetto al pivot si posizionano sul lato sinistro del pivot mentre gli elementi con un valore maggiore del pivot si spostano sul lato destro del pivot. La sezione sinistra è chiamata la partizione sinistra e la sezione destra è chiamata la partizione corretta.

**Differenza tra Quicksort e Merge Sort**

In primo luogo, la funzionalità è la principale differenza tra quicksort e merge sort. Quicksort ordina gli elementi confrontando ogni elemento con il pivot mentre l'unione di ordinamento divide l'array in due subarray (n / 2) ancora e ancora finché non viene lasciato un elemento.Inoltre, mentre quicksort è adatto per piccoli array, unisci lavori di ordinamento per qualsiasi tipo di array.  
Un'altra differenza tra quicksort e merge sort è che quicksort funziona più velocemente per i piccoli set di dati mentre l'unisci sort funziona a velocità costante per tutti i set di dati.

Perché posso non specificare la dimensione in un vettore, mentre nella matrice è obbligatorio specificare almeno le colonne?  
Il vettore è un array monodimensionale ed è allocato linearmente in memoria cio vuol dire che se lo si passa ad una funziona basta il nome e le parentesi quadre poichè il calcolatore, dato il primo inidirizzo sa calcolare i seguenti con vet+i .  
La matrice è un vettore di vettori quindi array bidimesionale, viene anch esso allocato linearmente quindi per passarlo nella funzione bisogna dargli il num di colonne perchè il calcolatore a causa del diverso metodo di accesso lo sa calcolare solo con esso.si fa vet+(i\*NCOLONNE)+j praticamente per ogni incremeto lo fa per il numero di colonne.

Cosa è e a cosa serve un file header?Per realizzare la Libreria

Se realizzassi una libreria, cosa metteresti nel file.h?  
I prototipi delle funzioni

Inizializzazione variabile con memset  
L'**inizializzazione** delle **variabili** nel linguaggio **C** è la fase preliminare della programmazione in cui assegnano i valori iniziali alle **variabili** precedentemente dichiarate  
**memset (str,'val da impostare ', n byte da impostare per quel valore N\*size of(tipo)).**  
Libreria string.h 3 parametri.

Quali sono le funzioni contenute in stdlib.h?  
Ordinamento, conversione = Malloc, Calloc , rand , atoi , atof.

FATTORIALE  
è piu semplice crearlo con approccio ricorsivo , è definita così:   
se n=0 fa 1 caso base  
se n>0 fai n\*funzione(n-1) passo induttivo

Come si usa rand?  
Bisogna prima inizializzare con srand(time(NULL))e poi n=rand() che da un num casuale da 0 a 3mila e qualc

COS è LA CODIFICA ASCII?

Codifica per i caratteri in cui si utilizzano solo 2 bit , 0 e 1, e si creano 2^8 combinazioni quindi 256 ogni carattere è una sequenza di 8 bit di 0 e 1.  
Rappresenta 128 caratteri. Serve per leggere facilmente da qualsiasi sist operativo

Permessi di scrittura

BISEZIONE  
è un metodo molto semplice per approssimare le soluz di un equazione (anche per calcolare zeri della funzione)

sscanf e sprintf, cosa sono e cosa fanno?

Sono funzioni che scrivonono/leggono su una stringa da stream e la convertono in un formato specifico o ritornano EOF se c'è stato qualche errore.  
Sprintf(formato,variabili) CONVERTONO VALORI IN SPECIFICO FORMATO

scanf su tastiera, fscanf da file , sscanf su una stringa

Come faccio a capire se un cast sia esplicito o implicito?  
Bisogna guardare com era e come diventa , perché nel cast implicito ci sono delle regole di “ priorità ” e quindi ad esempio int non puo mai diventare float , invece in caso esplicito l’operazione si forza quindi si può.

Float a int e int a float (esplicita se più preciso il tipo di destinazione), una delle 2 è implicita?  
int a float è esplicita e float a int implicita

Char a int è implicita o esplicita?  
Char a int esplicita.

Negato delle variabili booleane?

È il suo contrario cioè se la var è 0 il negato è 1, se la var è 1 il neg è 0 OPERAZIONE NOT

**Cambia qualcosa se r e c sono dimensioni statiche o si usa NMAX? (Nella lettura)**

Si perchè #define sostituisce prima della compilazione ed è una semplice sostituzione, si può modificare facilmente.   
Non appesantisce il codice poichè si passa solo dopo al compilatore.

Una dimensione statica inoltre dovrebbe essere richiamata nelle funzioni sempre.

DOMANDE ESAME 27

**Cosa cambia se fai il puntatore o il valore ?**